Contents

1 Ultimate goal

The ultimate goal of my machine learning work is to work towards automating fusion reactor optimization which sounds impossibly difficult but is theoretically possible. I am approaching this by working on a simpler machine (LAPD) and starting with simpler, feed-forward models and generative models later. The most difficult portion – collecting a good dataset – has already been done so hopefully this will all be straightforward.

2 Profile prediction (PP)

The profile prediction campaign has a few goals, categorized by physics and modeling. The primary physics goal of profile prediction is to determine the most important drivers of bulk profile changes, axial non-uniformity, and cross-field transport. A good model for profile prediction could be useful when setting the machine parameters for different data runs, depending on the objective of the run. On the modeling side, the goals are the following: determine how to train good models on LAPD data, determine if models should be separately trained for the run sets with different pressures (Feb 2023 vs Apr 2024), determine how useful explainability algorithms (e.g., SHAP) compare with directly dropping model inputs. These two goals would provide guidance on how to train and interpret subsequent models so it is important that insight is gained as early as possible.

Definitions in case of confusion:

- **Run set**: a collection of **dataruns** taken during a certain run period (usually a week) on the LAPD.
- **Datarun**: a set of back-to-back LAPD discharges executed in a smaller time period (hours) with a fixed machine configuration to collect probe data. Many of these are collected during a run week to compose a **run set**.
- **Training dataruns**: a particular set of **dataruns** used to train the ML model.
- **Dataset**: the data used in ML models.
- **Training set**: **examples** from a dataset used to train the ML model.
- **Validation set**: **examples** used during training to measure overfitting, typically created by holding out a fraction of shuffled **examples** from the training set.

- **Test set**: **dataruns** used to measure out-of-distribution (unseen-machine-configuration) performance.
- **Example**: a single item in an ML dataset. In this case, a single LAPD discharge.

3 PP1

3.1 Goals

The focus of phase 1 of the profile prediction campaign (labeled as PP1) are the following: learn the best practices for training models, learn how to tune the models for better performance, find the minimal set of inputs needed for accurate time-averaged profiles, create a baseline of linear or simple models, and to compare the insight provided by Shapley additive explanations (SHAP) with simply dropping out various inputs. All of this learning will be done on a very simple dataset where training is fast for rapid iteration. Lessons learned here can be applied to more complicated feedforward models in the future.

3.2 Data processing

The minimal dataset was created for profile prediction. The inputs have no recorded signals, only machine settings conveniently available to an LAPD user.

- Inputs: settings and probe position (11 items):
  - Source field (G)
  - Mirror field (G)
  - Midplane field (G)
  - Gas puff voltage (V)
  - Discharge voltage (V)
  - Gas puff duration (ms)
  - Probe position and rotation (xyz+θ, cm+rad)
  - Run set flag (training sets 03 and onwards)
  - Top gas puff flag (training set 04)
- Output: I_{sat} scaled to 1 mm^2 (as each probe may have a different area).

The I_{sat} signals were averaged from 10 to 20 ms. This range was chosen to avoid effects of ramp up, average-out plasma fluctuations, and provide a rough measure of average plasma density. Plasma profile evolution will be attempted in subsequent training phases and campaigns. An example plot of three I_{sat} time series near x=y=0 can be seen in fig. 1. Note that the I_{sat} time traces can vary significantly between axial (z) position and dataruns (machine settings).

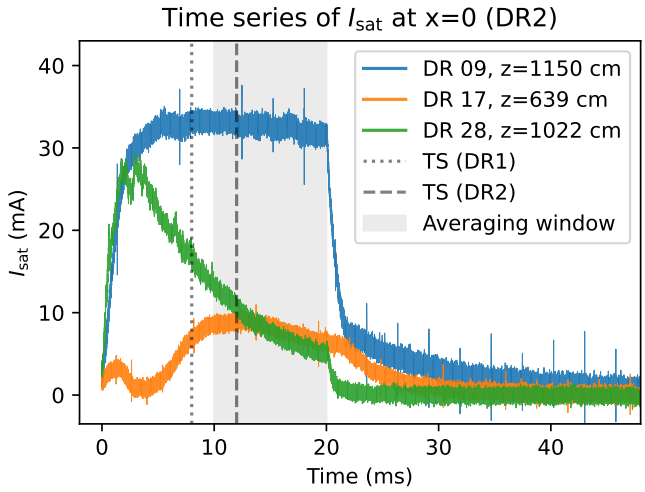


Figure 1: I_{sat} traces from a Langmuir probe from `DR2` dataruns 09, 17, and 28 at ports 17, 33, and 21, respectively. Shot 1 of 6 shown. Thomson scattering measurements are taken at 8 ms in `DR1` and 12ms in `DR2`. The window used for averaging is the filled region.

For the probe tip that was on the same shaft as the swept probe (in `DR2`), the signal was instead averaged over when the bias voltage on the swept tip was held constant at the lowest value. A 40 μs (250 sample) buffer was used after the sweep was turned off to minimize the impact of transient conditions. A comparison of the full trace and the trace with the swept portion excluded can be seen in fig. 2. Notably, the measured I_{sat} value does not attain a steady state before the discharge shuts off.

An I_{sat} value of 1 mA/mm^2 corresponds to 1-2 $\times$ 10^{12} cm^{-3} depending on Te.

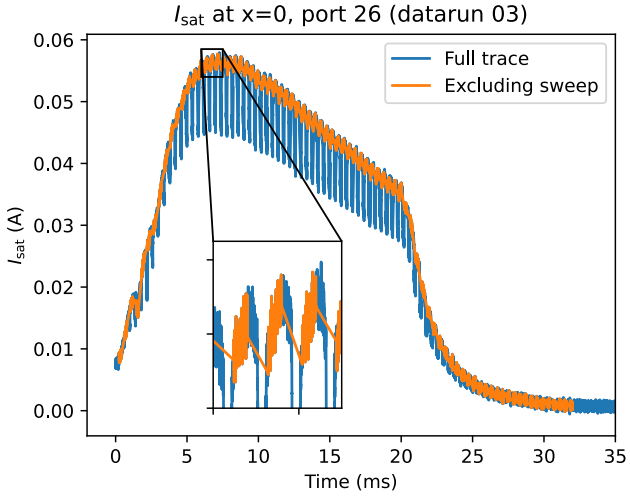


Figure 2: I_{sat} traces from the swept probe (port 26) from `DR2` datarun 03, shot 1 of 6. The orange curve is excluding times when a sweep is active on an opposing tip. Note that I_{sat} does not appear to fully recover so the averaged value may be a slight underestimate of the true I_{sat} (but this difference is ultimately insignificant).

I_{sat} profiles from `DR2` datarun 15, can be seen in fig. 3. This datarun was chosen because of the clean visual separation between the profiles. Each dot in the plot is a single training example. Note that each position may have some variation in the I_{sat} value recorded, but the model will only learn the expected value (generative models – future work – will learn the distribution). The difference in the profiles for the same machine settings but different axial positions implies the existence of a strong axial density gradient in the plasma. Deducing the machine parameters to create and remove these gradients is one of the goals of this campaign.

In total, 67 dataruns, each with different machine settings, are mixed or left out to form the training and test datasets. Most of these dataruns were randomly generated using Latin-hypercube sampling (LHS) from a set of machine configurations enumerated in the first five actuators in table 2.

3.2.1 `DR1` notes and peculiarities

For this run set, the recorded probe positions are used in the datasets. Some signals saturated either the digitizer or isolator early in the week run week; these shots were removed from the dataset. This run set had two I_{sat} probes on the machine at two axial positions: 895 and 830 cm (ports 25 and 27, respectively). These probes took measurements at many radial positions using probe movements in a line (i.e., always at $y = 0$), in miniplanes with three different y-coordinates, and in larger planes with many different y-coordinates. These movements are not evenly represented in the dataset and will be discussed later on in section 3.4. This run set also was in a high-neutral-pressure regime because only one turbo pump was functional.

3.2.2 `DR2` notes and peculiarities

Unlike in `DR1` , the recorded positions of the probes in this run set appeared to drift by up to ≈ 1 cm so the coordinates were instead generated from the defined motion list in each datarun. I do not know how the recorded coordinates drifted in the first place, or how that would even be possible. Three of the dataruns are overnight planes and the remainder are lines. In total, data were collected from four probes with one I_{sat} tip each, spaced axially along the machine at 1150, 1022, 863, and 639 cm (ports 17, 21, 26, and 33, respectively). These ports were chosen to have no overlap with ports from `DR1` so that the aggregated dataset would have greater axial diversity. This run set had all turbo pumps operational leading to lower neutral pressures for the same gas puff fueling settings.

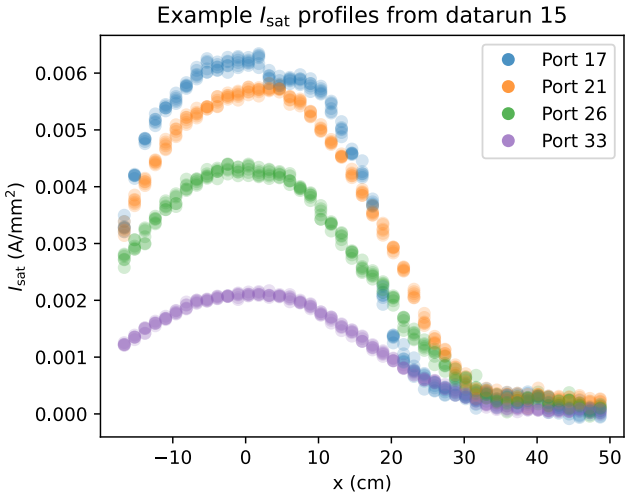


Figure 3: Examples of I_{sat} profiles from `DR2` datarun 15. Each point is a single training example. Note that each position has multiple examples.

3.2.3 Lack of I_{sat} calibration

The I_{sat} measurements of both dataruns are taken at face-value without calibrating with respect to the interferometer at port 20. Calibration is important because films or dust could build up on the probes, modifying the conductivity and thus the measured I_{sat}. This lack of calibration decreases confidence in the accuracy of our data. However, for `DR2` this error can be estimated by looking at dataruns with little (expected) axial variation. Namely, we consider the flat case with 5 ms gas puffing and high discharge voltage which leads to high-temperature, near-burnout conditions. The temperature was measured via Thomson scattering and is likely between 7.4 and 12.5 eV, densities around 10^{12} cm^{-3} and the low neutral density is apparent in the decreased broadband light emission recorded by the fast framing camera. The mean free path of electrons is then about a meter with collision frequencies around 2-3 MHz, so we expect the column to have small gradients axially. The profiles of the four probes from this run (`DR2` datarun 28) are seen in fig. 4. The profiles are very similar, as expected, though some strange behavior is seen at port 17. This similarity gives us confidence that the I_{sat} values from these probes are consistent with one another in cases where gradients are present.

This calibration process could not be performed for `DR1`, but those probes are only two ports ($\sim$ 64 cm) apart and seeing a 20% difference in I_{sat} around that point in the machine (port 25 and 27) is not surprising.

Calibrating I_{sat} measurements of `DR1` and `DR2` to each other is not possible in this case. The model may learn this discrepancy through the data run set flag. Note that test set performance does improve when the two run sets are mixed without the run set flag (fig. 9), so the discrepancy may not be that great.

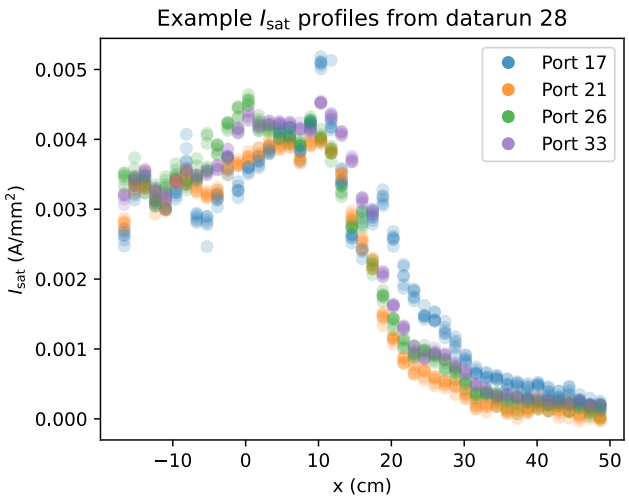


Figure 4: I_{sat} profiles from `DR2` datarun 28, which had $\approx$ 9 eV temperatures and electron mean free paths around a meter. Profiles remain similar along the machine, which suggests that the I_{sat} probes are relatively well-calibrated to each other for dataruns in `DR2`.

3.3 Training, validation, and test sets

Table 1: Summary of training and test datasets

Dataset	Training dataruns	Test dataruns Notes
01	62 dataruns	`DR1` : 08, 15, 33; `DR2` : 10, 31
02	59 dataruns	`DR1` : 08, 15, 23, 33; `DR2` : 02, 10, 19, 31
03	59 dataruns	7x random cross-validation, four from each run set. Added run set flag (scaled).
04	59 dataruns	7x random cross-validation, four from each run set. Added run set flag and top gas puff valve flag

A summary of training datasets used in this study can be seen in table 1. Initially, `DR1` dataruns 08, 15, 33 and `DR2` dataruns 10, 31 were left out to compose the test set. These dataruns were chosen to contain a variety of machine conditions, such as both lines and planes, varying gas puff lengths (in `DR2`), and a few latin-hypercube samples. This collection seemed somewhat small so dataruns `DR1` 23 and `DR2` 02, 19 were added. In dataset 03, a seven-way cross-validation set was also created without replacement starting with the test dataruns from training dataset 02.

3.4 Dataset diversity

Efforts were taken to maximum diversity of the aggregated dataset but the nature of the data collection process nonetheless introduces unwanted imbalances and regularities. A summary of the different configurations and the corresponding dataset percentage can be seen in table 2. A summary of the I_{sat} values and x-coordinate distributions, split by training and test set, can be seen in figs. 5 and 6, respectively.

Different line and plane configurations were used for `DR1` and `DR2` to better cover the position space. Measurements at random positions were not supported in the data acquisition software

and adding new run position configurations and verifying safe probe movement also takes time. A large assortment of position configurations also makes a conventional analysis more challenging the event that ML-based analyses do not perform as expected. Axial positions are largely permanent for the duration of a datarun because remove and reattaching a probe elsewhere on the machine is a process that can take several hours and requires stopping dataruns. Running line measurements is much faster than planes and thus allows a greater diversity of run configurations; planes are typically ran overnight because of the longer duration.

All of these factors lead to a dataset with only a handful of axial probe locations, many lines, and a few planes. The dataset is thus unbalanced with good radial ($y = 0$) coverage for most run configurations and detailed x-y plane measurements for only a handful of dataruns. This unbalance can be alleviated by forcing a hard prior of azimuthal symmetry on the ML model, but azimuthal symmetry is a poor assumption in my experience. The balance of different x-coordinates can be seen in fig. 6. Also note that in table 2 the percentage of shots with a nonzero y-coordinate is twice as great as the $y = 0$ shots, though the $y \neq 0$ dataruns are much less diverse.

The run week plan also had a significant effect on dataset balance. The initial dataruns in `DR1` were in configurations similar to standard, typical runs used on the LAPD. Other probes and diagnostic were being setup at the time these runs were taken, so some runs only had a single probe taking measurements. Many of these initial `DR1` dataruns had similar configurations so a trained model will be biased towards these typical LAPD configurations which is not necessarily bad if that is where the model will be typically used. `DR2` largely contained LHS-generated dataruns, so run conditions with lower neutral pressure will likely have better model performance over a larger region of machine configuration space.

Only a few dataruns with gas puff durations less than 38 ms were recorded. These dataruns were taken as an attempt to see mirror-related interchange instabilities in higher-temperature, less collisional regimes. Because of the small amount of dataruns in non-random machine configurations, good model performance on regimes similar to these is possible but not expected.

In general, as seen in fig. 5, the aggregated dataset is biased towards I_{sat} values below 0.01 A/mm^2. In addition, most of the higher-I_{sat} runs and values near zero occur in the lower neutral pressure regime that `DR2` covers which was not expected.

Ultimately, this dataset combining `DR1` and `DR2` covers a wide range of plasma conditions on the LAPD, but a larger, more diverse dataset (i.e., more of unique experimental configurations) may be needed for better interpolation and generalization.

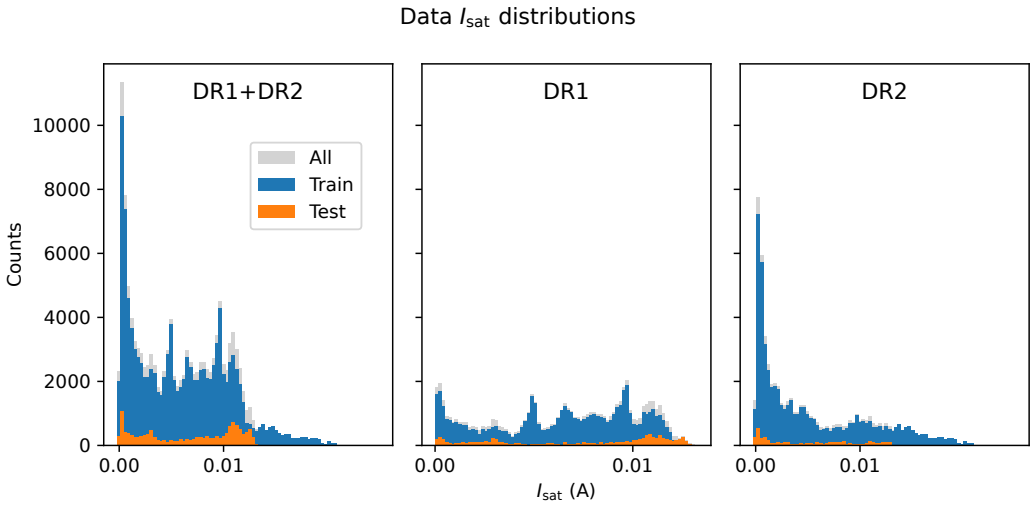


Figure 5: Distribution of I_{sat} signals. `DR1` appears to have a more uniform distribution than `DR2` does. Combining the two datasets yields a large peak near zero and dramatic flattening above 0.01 A/mm. From these histograms we expect or model to be biased towards fitting lower I_{sat} values better, and to perform badly in cases with high I_{sat} values.

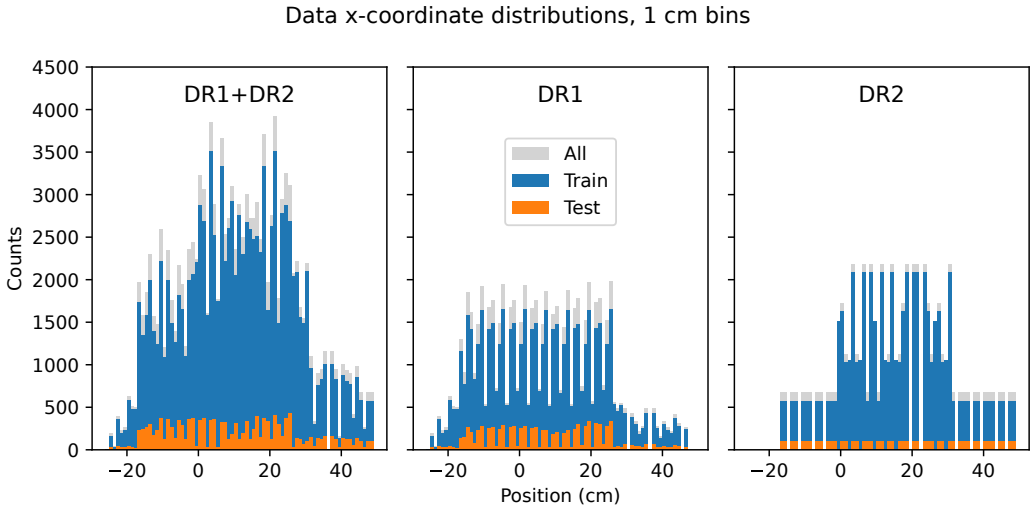


Figure 6: Distribution of the x-coordinate in the profiles. The increase in data points between roughly x $\approx$ 0 to 30 cm is from planes instead of lines. Based on this distribution, the performance of the model is expected to be biased towards this central area.

Table 2: Data breakdown by class and dataset (percent)

B source (G)				B mirror (G)				B midplane (G)			
	Train	Test	All		Train	Test	All		Train	Test	All
500	4.77	0	4.29	250	4.30	8.41	4.72	250	8.25	21.01	9.55
750	3.34	12.61	4.29	500	30.49	8.41	28.23	500	43.80	8.41	40.19
1000	43.13	78.99	46.78	750	6.68	16.81	7.72	750	6.62	52.19	11.27
1250	12.59	0	11.30	1000	28.85	57.97	31.82	1000	26.36	5.78	24.26
1500	19.23	0	17.27	1250	3.34	4.20	3.43	1250	9.24	0	8.30
1750	1.91	0	1.71	1500	26.34	4.20	24.08	1500	5.73	12.61	6.43
2000	15.03	8.41	14.35								

Gas puff voltage (V)				Discharge voltage (V)				Axial probe position (cm)			
70	12.11	16.81	12.59	70	12.22	8.41	11.83	639	12.48	8.41	12.06
75	6.68	0	6.00	80	5.25	0	4.72	828	17.07	36.28	19.03
80	11.46	8.41	11.15	90	2.86	8.41	3.43	859	12.48	8.41	12.06
82	41.49	57.97	43.17	100	3.34	8.41	3.86	895	33.01	30.10	32.71
85	14.13	0	12.69	110	8.77	0	7.87	1017	12.48	8.41	12.06
90	14.13	16.81	14.40	112	20.62	0	18.52	1145	12.48	8.41	12.06
				120	3.82	8.41	4.29				
				130	0.95	0	0.86				
				140	2.86	8.41	3.43				
				150	39.30	57.97	41.20				

Gas puff duration (ms)				Vertical probe position (cm)			
38	94.27	91.59	94.00	≈ 0	36.26	46.08	37.26
< 38	5.73	8.41	6.00	$\neq 0$	63.74	53.92	62.74

4 PP1 benchmarks

4.1 Baseline linear-like models (`DR2`)

A linear model obviously cannot fit the dataset (see the nonlinear shape in fig. 3). However, a simple (and mostly linear) model can provide a performance baseline to help spot bugs when training more complex models. Since the x- and y- profiles have a approximate tanh shape, a feature is added at the linear model input stage for the x and y coordinates: $x_{\texttt{tanh}} = c \cdot \tanh\left(|x + s| \cdot a + b\right)$ where s, a, b, c are trainable parameters (independent for each coordinate; c is superfluous). This function was chosen to give the linear model the capability of expressing tanh-like curves. The performance of the linear model on `DR2` data, with and without the tanh features, can be seen in fig. 7. This baseline linear-like model reaches a training and validation loss of around 0.011, with the RMSE $= \sqrt{\text{loss}} \sim 0.1$. The linear-only model is marginally worse with losses at around 0.014.

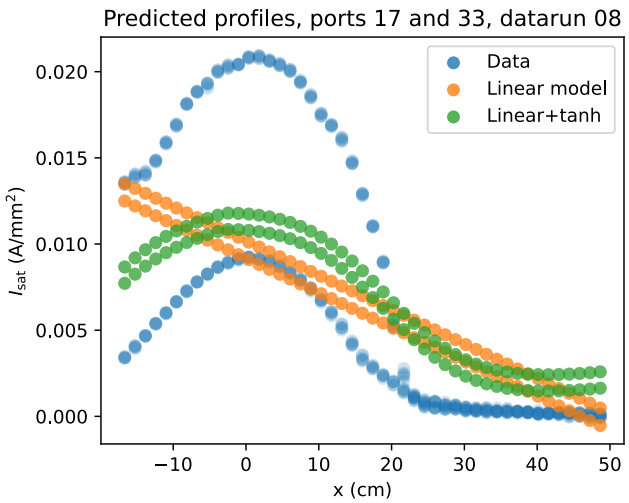


Figure 7: I_{sat} profiles and predictions for ports 17 and 33 based on inputs from `DR2` datarun 08 using a liner and linear+tanh model. Each point is a single prediction. Datarun 08 was chosen for its representative performance; ports 17 and 33 were chosen to demonstrate the maximal axial variation (across 511 cm). These models fail to describe the data accurately.

This feature engineering-like approach can continue. For example, the width of the profile is largely controlled by magnetic field configuration of the device, particularly by $\sqrt{B_{\text{midplane}}/B_{\text{source}}}$. This behavior can be added to this model, either as a new feature or as a custom relationship in the model. Note that, as seen in fig. 7, the width of the profile also depends on the axial coordinate. Combined with other coordinates and actuators, like discharge voltage and gas pressure, the number of possible features or function space grows combinatorially, making this custom fitting process difficult and tedious to design and test by hand. The obvious solution would be to use symbolic regression or fitting to a function library which may be ideal methods if simple profile prediction were the final goal. However, we are ultimately interested inferring trends in a much more complex input space where neural networks are more flexible and accurate. If NNs do face generalization issues, symbolic regression or a SINDy-like approach can used instead, albeit with limited applications. Symbolic methods are appealing because the fits are simple. However, even though a simple equation may fit the data well, it does not necessarily provide insight or relate to the underlying physics; using a freeform fitting function like a neural network is more appropriate in this use case.

4.2 Pipeline validation

Andrej Karpathy's advice for training neural networks [3] was used as a template for verifying the training procedure used in this project.

The data fed into the model immediately before the forward pass (and subsequent backpropagation) was stored and verified: the data are correctly randomly shuffled in each batch. Each epoch contains the same random shot order. Many subtle bugs can occur in the data pipeline so verification is important and ensures correctness.

A simple dense model (4 layers, 512 wide with one output; 794113 parameters, tanh activations) was trained on a zeroed-out input as a baseline for determining that the model is learning anything at all. This process yields a validation loss of 0.036, which is 3-4x worse than a simple linear model. The model is also overfit on a single batch (128 examples) of training data to make sure that

training progresses as expected. A deep double descent is observed as expected [5, 7]. Training on a batch of 8 examples reaches ≈ 0 training loss after 50 steps. Plots of the train and validation losses can be seen in fig. 8. The 128-example model has a similar validation loss to the linear-like model and yields similarly-shaped profiles.

Multiple models were trained with varying depths and widths to verify that training loss decreases with increased model capacity. Doubling the layer width from 512 to 1024 moderately decreases the training loss; doubling the depth of the network from 4 to 8 layers has a larger impact. Increasing the width further to 2048 and depth to 12 layers has a dramatic impact on training loss, so this model and dataset are behaving nominally.

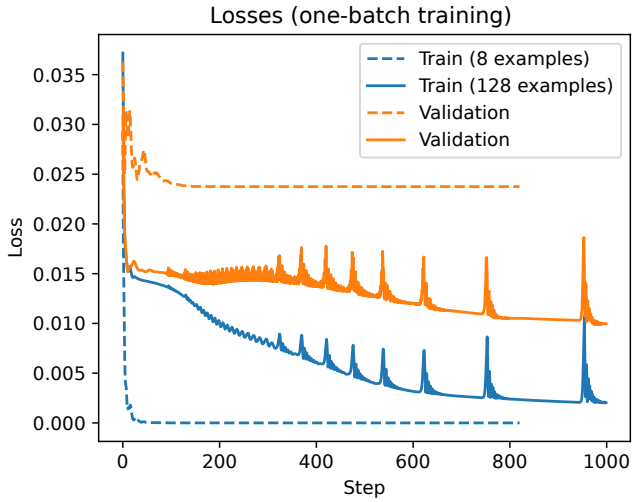


Figure 8: Training and validation losses when overfitting the model. A deep double descent in the validation losses is observed when fitting a single batch of 128 examples. The 8-example batch hits near-zero loss after 50 steps. This process verifies our training process is functioning as expected. The spikes are from exploding gradients which can be mitigated by clipping the gradients.

5 PP1 training, tuning, and generalization

The results in this section are largely presented chronologically. The goal in this section is to see which techniques work best on simpler datasets and simpler models before while they are cheap to train.

5.1 Training details

The full training dataset has 118,131 examples. The loss used to train the model was simply the mean-squared error (MSE):

$$\mathcal{L}_{\text{MSE}} = \frac{1}{m}\sum_{i=1}^{m}\left(f\left(x_i\right) - y_i\right)^2 \tag{1}$$

where x_i is the inputs for the ith example, y_i is the ground truth, m is the number of examples in a given batch, and f is the neural network. During training, overfitting was tracked using a traditional 80-20 train-validation random split. A dense neural network, 4 layers deep and 512 units wide, was trained with AdamW using a learning rate of 3×10^{-4}. Leaky ReLU activations were used instead

of tanh to avoid vanishing gradients and adaptive gradient clipping [8] (cutting gradients norms above the 90th percentile of recent norms) was used to mitigate exploding gradients.

Test set loss was evaluated during training on three out of four test dataruns from each run set (DR1 runs 08, 15, and 33; DR2 runs 02, 10, and 31), leaving two for post-training model evaluation (DR1 run 23; DR2 run 19). Evaluating the test set while training may initially appear as bad practice, but there must be both sufficient diversity in the training set to learn broader trends and also be sufficient diversity in the test set to accurately evaluate out-of-distribution performance. A large variance in test loss is observed between the held-out test runs which was picked explicitly for its diversity. The two held-out test set runs are left for evaluation after multiple models have been trained to verify that a real decrease in train-time test loss. Since no model had anomalous test loss this holdout test set of two shots has not been evaluated.

Train and validation losses were typically around $\approx 6 \times 10^{-5}$. No overfitting was observed, so we conclude that early stopping and sheer dataset size are sufficient so that no additional regularization is required.

5.2 Effect of dataset diversity

The goal of this PP1 campaign is to predict profiles in a wide range of plasma conditions, so number of different dataruns is used as a proxy for dataset diversity. This goal requires good model performance on unseen dataruns. Including the test set, there are 67 dataruns which may not be sufficient for deep learning-based models. In this section the effect of model performance on number of different dataruns in the training set is evaluated. The main results of this quick study in diversity can be seen in fig 9. An equal amount of dataruns from each run set were randomly and iteratively removed from the training set. Results when changing training set diversity are roughly what was anticipated: model performance on the test set decreased with the decrease in diversity.

In fig. 9, the test losses were averaged over batch iterations 4000 to 4500 which was the tail end of the 9-datarun model. The loss leveled out quite quickly into training so the particular iteration range the average was performed over did not make a significant difference. Multiple models with different seeds were trained for the full-training-set model to measure the performance variance over model parameters. The DR1 -only dataset was evaluated on the DR1 -only test set, and likewise with DR2 . The cross-run test set losses were incredibly high, near or above the zero-input baseline of 3.6×10^{-2} and so were not included in this plot. The "best" model was a large 12-deep 2048-wide dense network trained on the full training dataset, evaluated at 30 epochs. Longer training or larger models may yield better test set results, but will likely not come close to the training and validation losses which are on the order of 10^{-5}.

One unexpected result is that models trained on a single run set appear to perform worse when evaluated on the test set (from that particular run set) compared to the mixed training set of the same size (green arrows vs blue x). Skepticism is warranted, however, because the variance over model parameters may be able to account for most, if not all, of the performance difference as suggested by the full-set ensemble. The variance caused by random datarun selection may also account for some of the difference; this variance was not measured and could be considerable. The effect of varying the training and test datasets will be discussed in section 6.3.

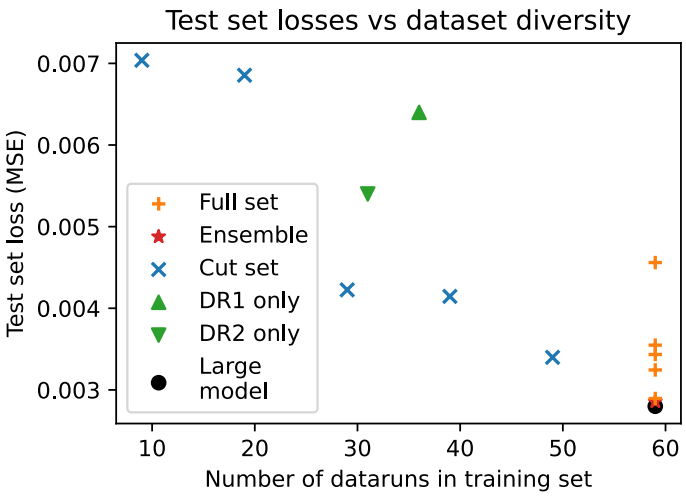


Figure 9: Several models were trained with dataruns cut out, decreasing the diversity of the training set. The test set was kept fixed. Loss on the test set monotonically increases with decreasing diversity.

All of this analysis was performed using one training set and one test set. Doing cross-validation using different combinations of training test dataruns would improve confidence in this analysis, but it's pretty clear that a reduction in dataset diversity decreases test loss performance regardless.

5.3 Ensembles for uncertainty quantification

As seen in fig. 9, ensembles are strong. Five networks (4 layers, 512 wide) with different initialization seeds were trained. The mean of the MSE from each individual prediction in the ensemble is fairly high, around 0.004. When instead the average *prediction* of the ensemble is taken and the error is calculated, the test MSE drops to below 0.003, which is nearly identical to the performance of a single larger model but with a fraction of the computational cost. This result indicates that, where possible, ensembles can provide much better test set performance with no other changes.

The predictions from ensembles can also be interpreted as a probabilistic measure of uncertainty [4]. However, the standard deviation of the single-point predictions from this dataset does not appear do not correlate very with the error in the prediction, suggesting that this uncertainty metric may not be useful on single point estimates.

The dataset cross-validation method I'm using is called "bagging" in ensemble literature. I could use "boosting" instead to weight the examples with the largest error, but that requires sequential training (the next model trained relies on the previous one). Bagging typically decreases variance but does not affect bias much; boosting improves the bias but the variance of the prediction does not change much. The cross-validation process gives insight into the variance of our model over the data it is trained on, which is separate from the variance over weights caused by the stochastic training process.

5.4 Improving test set performance with a "run set" flag

Run sets `DR1` and `DR2` were taken roughly 14 months apart so the machine state was different from the two runs. In `DR1`, only one turbo pump was operating leading to much higher neutral pressures than in the `DR2` run set. A new flag (mean-centered and scaled) was added to the inputs

indicating which run set each shot belongs to with the idea that the model would recognize the different behavior of the run sets and perform better on unseen dataruns. An ensemble prediction with this run set flag brings the test set loss down to 0.0019. The test set minimum between 5k and 10k batch iterations (roughly 6.8 to 13.5 epochs) indicates that a more biased model may perform better, e.g., weight decay may improve test set performance at the expense of train and validation performance. Significant weight decay helps with model calibration as demonstrated in Guo et al. 2017 [2].

5.5 Summary of model performance so far

A summary of the test set losses of varying datasets and ensembles can be seen in tab. 3. These results are only a rough comparison; the variability in prediciton performance with respect to model weights can be quite substantial, so losses without a uncertainty interval should be taken with a grain of salt. The number of training epochs varied between the comparisons in the table but that has at most a $\approx 10\%$ performance boost to the models only trained on `DR1` and `DR2`. Nevertheless some conclusions can be made here: models perform better with diverse data (also obvious from fig. 9), models perform better with more informative inputs, ensembles can give a substantial performance boost and an uncertainty metric, and performance improves with larger models and longer training.

Table 3: Summary of test set losses for different training data and ensembles

Model	Test set loss
`DR1` only	6.4×10^{-3}
`DR2` only	5.4×10^{-3}
Full set, large model	2.8×10^{-3}
Full set average	$3.6 \pm 0.56 \times 10^{-3}$
Full set ensemble	$2.9 \pm 1.1 \times 10^{-3}$
"Run set" flag average	$2.1 \pm 0.15 \times 10^{-3}$
"Run set" flag ensemble	$1.9 \pm 0.64 \times 10^{-3}$

5.6 Predicting a Gaussian distribution instead of a single value

Instead of predicting a single point, the model can instead output a mean μ and variance σ^2 using the negative-log likelihood (NLL) loss [6, 4]:

$$\mathcal{L}_{\text{NLL}} = \frac{1}{2}\left(\log \sigma_i^2(\mathbf{x}_n) + \frac{\left(\mu_i(\mathbf{x}_n) - y_n\right)^2}{\sigma_i^2(\mathbf{x}_n)}\right) \tag{2}$$

where i is the a model in the ensemble and n is an example. The mean over examples is implicit. This loss assumes the prediction – the likelihood of y given input $\mathbf{x}$: $p(y|\mathbf{x})$ – follows a Gaussian distribution. Treating each prediction as an independent random variable (considering each model in the ensemble is sampled from some weight distribution $\theta \sim p(\theta|\mathbf{x}, y)$) and finding the mean of the random variables yields a normal distribution with mean $\mu_*(\mathbf{x}) = \langle \mu_i(\mathbf{x}) \rangle$ and variance $\sigma_*^2 = \langle \sigma_i^2(\mathbf{x}) + \mu_i^2(\mathbf{x}) \rangle - \mu_*^2(\mathbf{x})$ where $\langle\rangle$ indicates an average over the ensemble.

The ensemble predictive uncertainty can be broken down into the aleatoric and epistemic components [10]: the aleatoric uncertainty is $\langle \sigma_i^2(\mathbf{x}) \rangle$ and the epistemic uncertainty is $\langle \mu_i^2(\mathbf{x}) \rangle - \mu_*^2(\mathbf{x}) = \text{Var}[\mu_i(\mathbf{x})]$. The intuition behind these uncertainties is that the random fluctuations in the recorded data are captured in the variance of a single network, σ_i^2, so the average of these variances represents that sort of randomness present. If the choice of model were to make a significant difference,

we'd expect the predicted mean for a single model, μ_i, to fluctuation quite a bit, which is captured by $\mathrm{Var}[\mu_i(\mathbf{x})]$.

The primary advantage of this method of predicting a distribution is this breakdown of uncertainty. Previous experiments indicated that simple ensemble-based uncertainty was not useful. The ratio of the two may be useful for identifying if the prediction is too out-of-distribution phil: should be rephrased . As seen in fig. for the test dataset, the epistemic uncertainty was much higher than the aleatoric – the opposite was true when the uncertainty was determined from the training set.

It's unclear to what extent the choice of test set is affecting the conclusions of this ensemble-based analysis. A cross-validation study is performed in section 6.3.

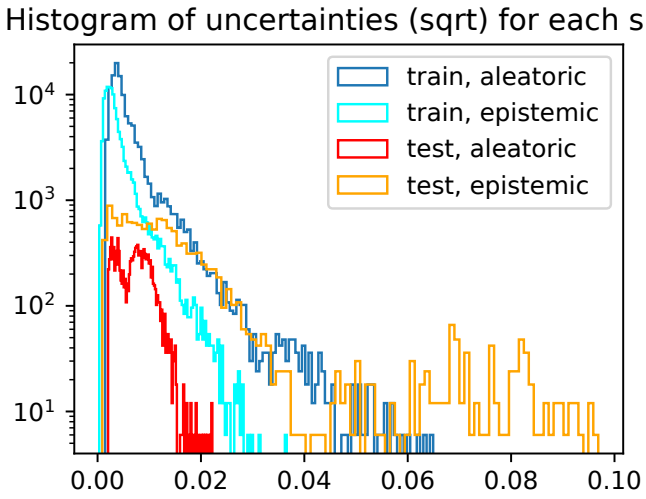


Figure 10: Breakdown of uncertainty into aleatoric and epistemic for the training and test datasets. The test dataset shows much higher epistemic uncertainty and much lower aleatoric uncertainty, on average.

Ideally, the true value should fall within the errors indicated by the prediction. An appropriate metric for determining model performance is the z-score, which is how far the prediction falls from the true value scaled by the standard deviation. Note that the z-score squared is one part of $\mathcal{L}_{\mathrm{NLL}}$ (eq. 2). The histogram of z-scores for predictions on the training and test sets can be seen in fig. 11.

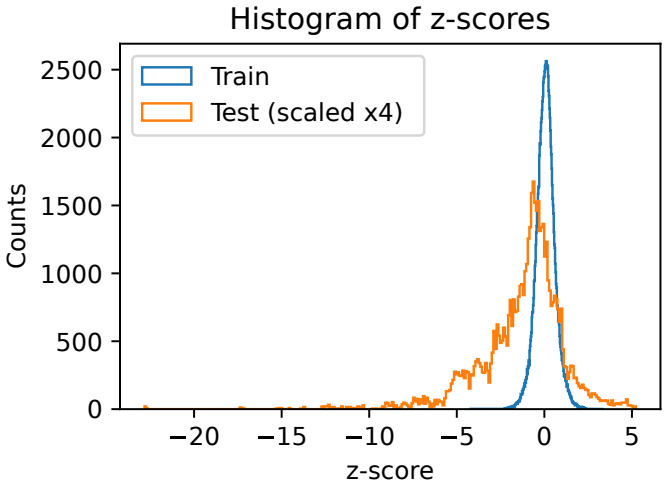


Figure 11: Z-scores of predictions of shots in the training set and shots in the test set when using a model that assumes a Gaussian likelihood distribution. Test set counts are scaled 4x for clarity. The model ensemble is clearly poorly calibrated.

One thing to note: the distributions over the ensemble and individual predictions are assumed to be Gaussian, so the calculated z-scores should also be Gaussian. However, this is clearly not the case in fig. 11 so the underlying normality assumptions are incorrect. For the training set, 92.8%, 99.7%, and 99.9% of the samples fall within 1, 2, or 3 standard deviations. And for the test set, 45.4%, 67.5%, and 80.7%, respectively. This model ensemble has poor test set performance and the uncertainty is not well-calibrated.

5.7 Weight decay for improving test set calibration

One of the goals is to make good predictive performance on the test set. This prediction is well calibrated if the true value reasonably falls within the distribution specified by the model. Increased weight decay can lead to better model calibration [2]. Many weight decay values are scanned to determine the best weight decay coefficient determined by the distribution of z-scores of the training and test examples. A comparison of many different weight decay coefficients can be seen in fig. 12; 20 was determined to be a good weight decay value. The "knee" value of 30 was not chosen because the distribution of z-scores looked odd. The training and test z-score distribution can be seen in fig. 13. Note that weight decay (coefficient λ) is implemented in `AdamW` as $\theta_t = \theta_{t-1} - \gamma\lambda\ \theta_{t-1}$ so learning would be impossible above the reciprocal of the learning rate of $\gamma = 3 \times 10^{-4}$.

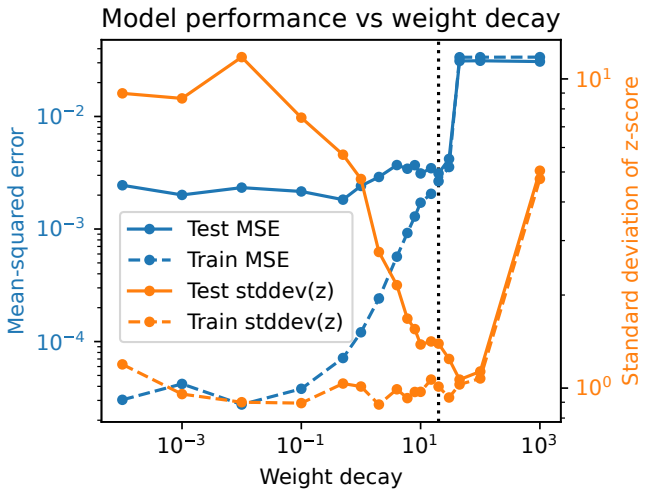


Figure 12: The MSE and standard deviation of z-scores over the predictions from the training and validation datasets. Models were trained on 17 different weight decays, from 0.0001 to 1000, shown as dots on the plot, were tested. A weight decay of 20 led to a good balance of lower training and test MSE but with low z-scores in predictions.

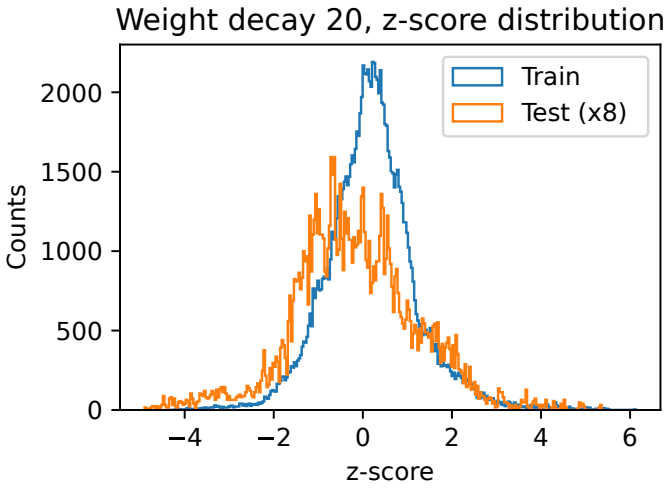


Figure 13: The distribution of z-scores for a model trained with a weight decay coefficient of 20.

Despite a well-calibrated test set error, the predictions on the training and test sets are not very good. An MSE of $\approx 3 \times 10^{-3}$ may sound decent (RMSE of $\approx 5\%$) but the error is weighted by the low I_{sat} values at the edge of the plasma (see fig. 5) which are relatively easy to predict correctly.

Using a weight decay coefficient of 20, the percentage of test examples enclosed bounded by 1, 2, and 3 standard deviations from the mean are 55%, 87%, and 95% respectively, which are much closer to what is expected from a normal distribution (68%, 95%, and 99.7%). Perhaps KL-divergence of the z-score with a normal distribution would be a better metric to describe the calibration of the network on the test dataset.

The test set MSE and calibration does not describe all predictive behavior, however. Although

the test set MSE is lower, the model has difficult producing the correct shape on the negative side of the `x`-axis as seen in fig 14. This behavior could be because of fewer examples on the negative portion of the `x`-axis, or the network simply lacks the capacity required to model the appropriate profile.

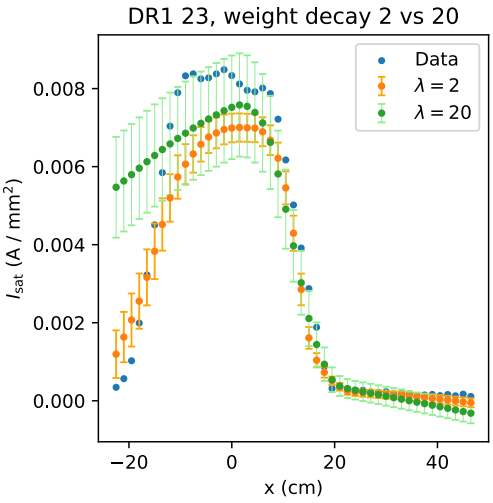


Figure 14: Comparison of profile predictions with uncertainties for a model with weight decay coefficient $\lambda = 2$ and $\lambda = 20$. The $\lambda = 20$ model is better calibrated but at the expense poor shape predictions. `DR1` datarun 23 is a representative example.

Interestingly, model the ensemble of models trained with the NLL loss is about as well calibrated as a model with a weight decay coefficient of around 2 to 8, which suggests that NLL-loss ensembles with additional weight decay may make well-calibrated, accurate predictions. Weight decay does appear to help with suppressing extreme predictions.

5.8 Assuming azimuthal plasma symmetry

The bias of the model can also be increased by assuming azimuthal symmetry of the plasma by just supplying the radial coordinate instead of the x-y-rotation coordinates. This effectively places a hard prior on the model. The distribution of this coordinate can be seen in fig. 15. This simplifies the learning problem by reducing the dimensionality of the input space by two (because the `y` and the probe angle are removed), puts planar dataruns on similar footing with lines, and increases the diversity of the coordinate. This azimuthal symmetry assumption can lead to peaked profiles because there is no longer continuity of I_{sat} profile gradient implicit in the dataset.

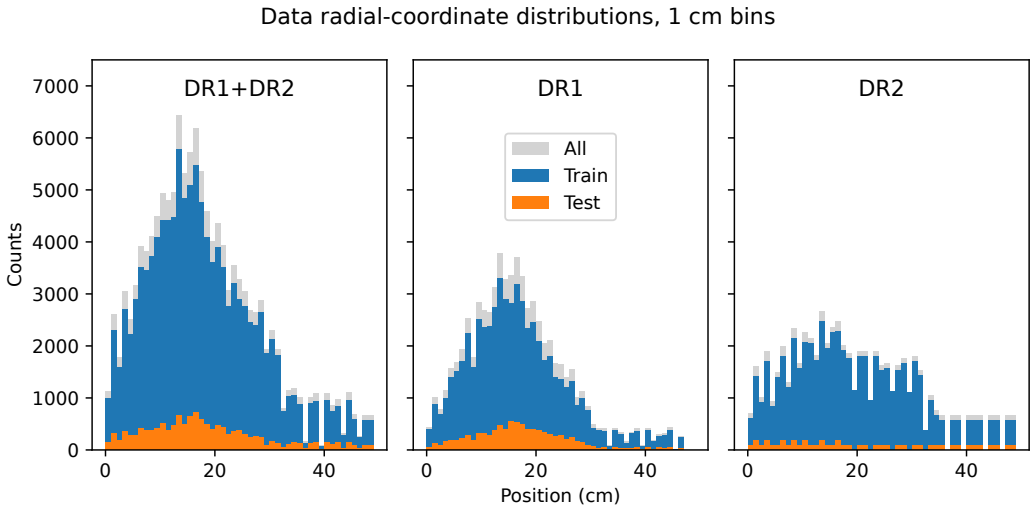


Figure 15: The distribution of the radial coordinate $r = \sqrt{x^2 + y^2}$. This coordinate is smoother over the input range than the `x` coordinate seen in fig. 6 and reduces the input dimensionality by two.

Performing a weight decay scan using this coordinate replacement leads to a similar optimal weight decay coefficient as seen in fig. 16. Notably, across all weight decay coefficients, the test dataset performance of this model is at least twice the value from the cartesian coordinate model.

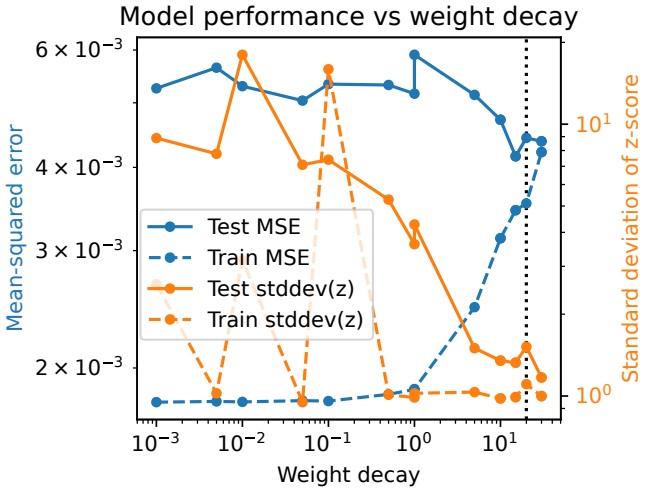


Figure 16: Using only the radial coordinate, the test and train MSE and z-score standard deviations are plotted as the weight decay coefficient is varied. Optimal weight decay appears to be around 20-30 which is consistent when the model is instead trained with the `x` and `y` coordinates.

5.9 Adding flag for top gas puff valve

For the first 9 dataruns of `DR2`, the gas puff valve on top of the machine was injecting gas into the chamber. This was disabled for dataruns 10 and onwards because of the unstable controller. A new dataset was created, `PP1` 04, that adds a flag indicating whether this valve was active. Training

with this model yields a test set MSE of 1.81×10^{-3}, which is close to the NLL ensemble value of 1.79×10^{-3}. Although it's not clear whether or not it outperforms the ensemble, the 04 dataset does outperform 6 out of 7 models from the ensemble, so it's likely that this flag is beneficial. It does not appear to benefit model calibration, but this new dataset will be used going forward.

5.10 Lessons learned so far

The exploration in this section was primarily to develop intuition and test methods for training networks on relatively cheaper models. Below is a list (in no particular order) of lessons learned during this process.

- Large amounts of weight decay helps with model calibration, but may not capture the shape accurately.
- Training with NLL loss enables breaking down the uncertainty into aleatoric and epistemic uncertainties, but those may be untrustworthy.
- Using a NLL loss improves test set performance.
- Forcing azimuthal asymmetry does not work too well.
- Ensembles both have better prediction error and are better calibrated than a single model.
- Providing more information to the model in the input seems to help a little.
- Training for longer improves prediction error.
- Large models have better prediction error.
- Increased weight decay appears to help with training stability.

Much more with these models could be tuned, but I've decided to focus on what seems most important. There's also a tendency for people to underexplore and overexploit in these sort of situations, but I do want to do due diligence, but it's important not to get stuck in this local minima of model tuning. It'll probably be better to try wilder things.

5.11 Training instability

While training some models, the training process encounters some instability. Although adaptive gradient clipping is used by clipping based on the 90th percentile of the last 1000 steps) that is not aggressive enough. Occasionally the validation MSE will spike, leading to a spike in the test set MSE as well. Though, the test set MSE can spike with no seen cause in the training or validation MSE. At the end of training, when the model is evaluated, some of the models can still be in this unstable state, leading to very poor predictions. These misbehaving models are removed from the ensemble, sometimes leading to much worse ensemble-based statistics than would otherwise be desired. This instability may also be the source of the large spikes in training standard deviation seen in the weight-decay model-performance plot in fig. 16.

6 Developing a useful model

Taking from what we have learned from previous experiments (see section 5.10), we train 5-network ensembles with larger individual networks (8 layers, 1024 units each) for 500 epochs with a variety of weight decay coefficients (0.01, 0.1, 1.0 and 10.0). These models are trained on dataset 04 (see table 1) which has the run set and top-gas-puff flags and no azimuthal symmetry assumption.

6.1 Scanning weight decay

To find a well calibrated model the ensembles are trained with a variety of weight decay coefficients. Fig. 17 plots the MSE and standard deviation of the z-score as a function of the weight decay coefficient and indicates that optimal calibration is between $\lambda = 1$ and $\lambda = 10$. The test set error increases more than is desired before the total ensemble variance is sufficiently large. This result may indicate another method is needed for model calibration outside of ensembles and weight decay, such as a β-NLL loss function where the NLL loss (eq. 2) for each example is scaled by a factor of $\sigma^{2\beta}$. The test set standard deviation is consistently higher than the training set. The distribution of the z-scores for this ensemble is seen in fig. 18. A comparison of the predictions of these ensembles with different weight decay coefficients can be seen in fig. 19. The largest weight decay ($\lambda = 10$) leads to the greatest uncertainty, but struggles to accurately model the shape of the profile. Lower weight decay leads to lower uncertainity, but better accuracy.

The z-score histogram (fig. 18) shows that the model has a tendency to underestimate the I_{sat} magnitude, but not overestimate. The lackluster performance on the test set on these underestimated points indicates that the uncertainty metrics used in this model (ensembles and NLL loss) may not be all that useful. The breakdown in to aleatoric and epistemic uncertainty, seen in the profile predictions (fig. 19), appears to correlate with each other and do not seem to provide any particular insight into the sources of uncertainty. This breakdown may not be useful. The total variance – ensemble and NLL – is useful, however.

The $\lambda = 0.01, 1.0$ and 10.0 models suffered from instability during training, so stricter gradient clipping method is necessary when training at lower weight decay coefficients. For this weight decay scan, 3, 0, 2, and 1 runs needed to be removed from the 5-model ensemble for $\lambda = 0.01$, 0.1, 1, 10, respectively.

We expect the aleatoric uncertainty to be small compared to the absolute I_{sat} value because the I_{sat} value does not tend to fluctuate very much shot-to-shot.

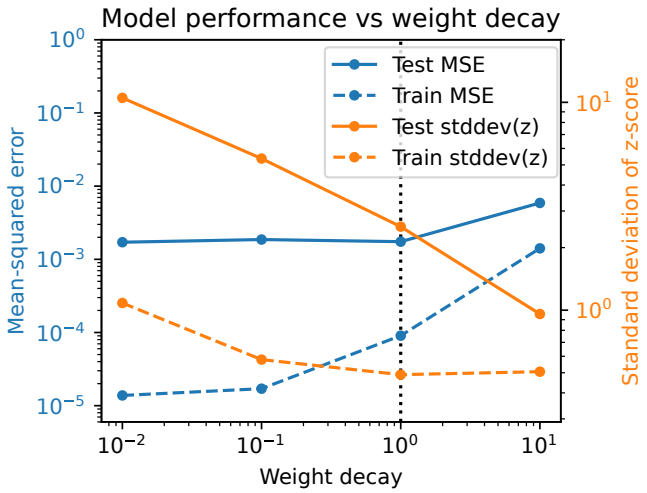


Figure 17: Performance of the model ensemble on the test and training datasets for four different weight decay coefficients. Optimal calibration appears to occur between $\lambda = 1$ and $\lambda = 10$.

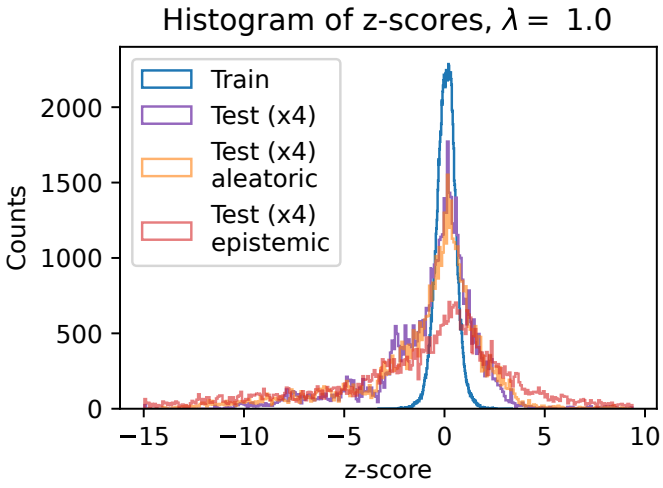


Figure 18: Z-scores of predictions (using the total variance) of an ensemble of 3 larger models trained with a weight decay of $\lambda = 1$.

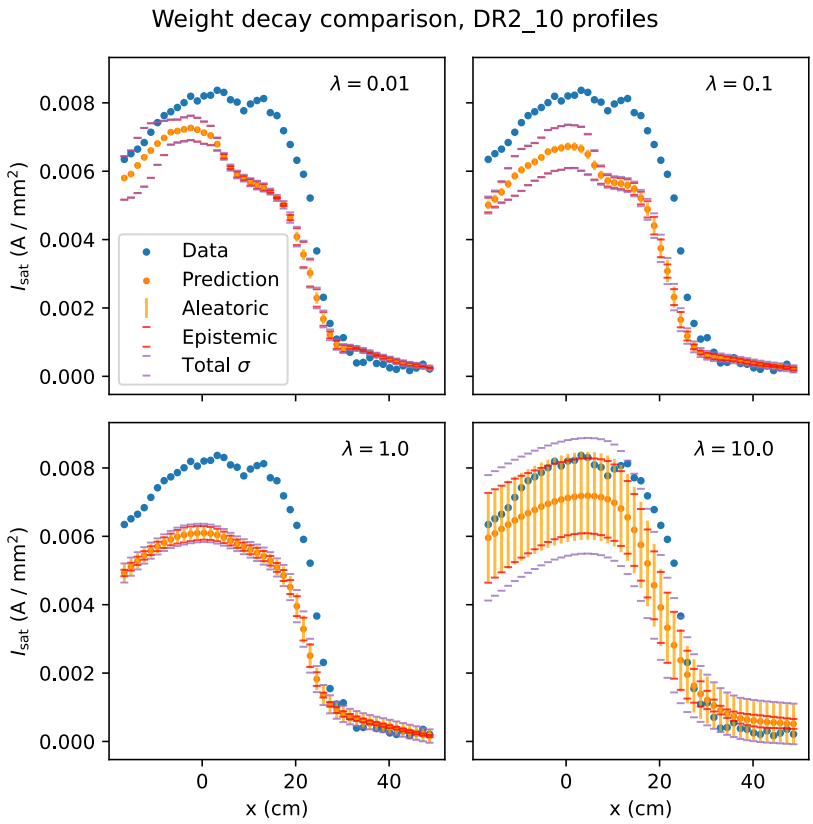


Figure 19: Comparison of the four different weight decay coefficients on test set `DR2` datarun 10.

6.2 Characterizing uncertainty

We have two methods of uncertainty quantification: ensembles of models and predicting the mean and variance of a Gaussian distribution by structure of the loss function. In order to be useful, these uncertainty metrics must give large uncertainties when the error is likely to be large, such as predictions outside the range of training data or in regions (in input space) of large variability (such as the gradient region along the `x` coordinate). Fig. 20 shows the uncertainty in predictions for ensembles with four different weight decay coefficients. At $\lambda = 1$ or above, the uncertainty does not appreciably increase compared to interior points when predicting beyond the training data limits. Uncertainty predictions in $\lambda = 10$ are entirely unreliable. For $\lambda = 1$ the gradient region has high aleatoric uncertainty, which is expected (and good) because slight changes in position can lead to rapid changes in I_{sat} magnitude.

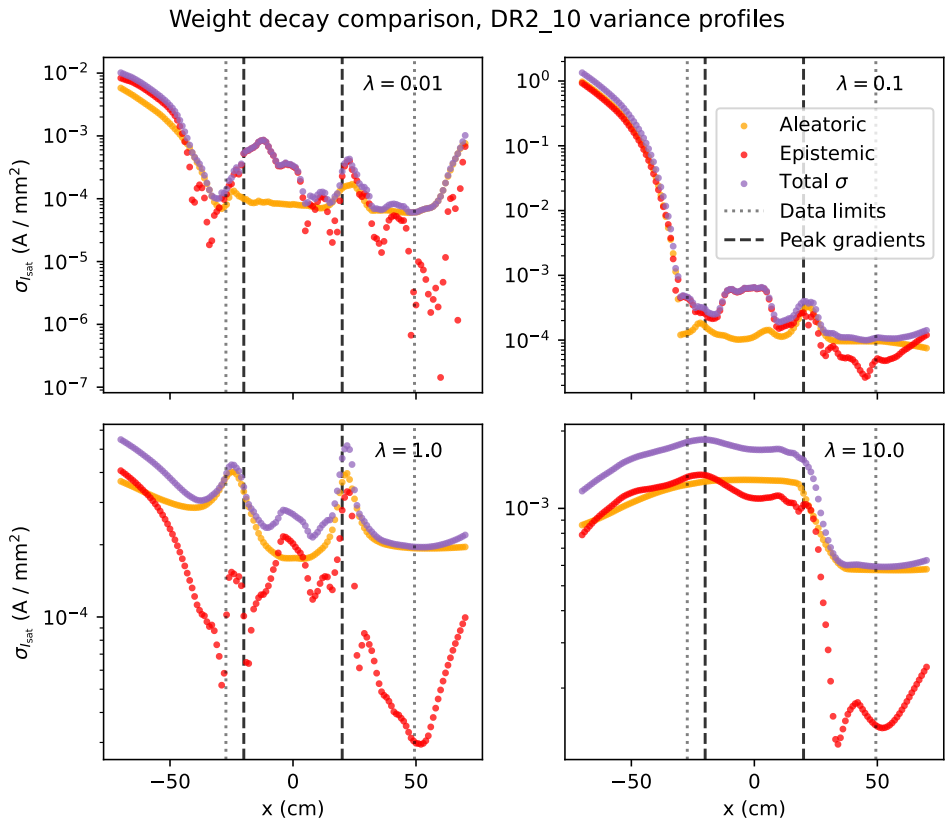


Figure 20: The predicted uncertainty for four different weight decay coefficients for each ensemble on test set DR2 datarun 10. Note that the pridictions extend outside of the training data limits, and also into the unphysical region past a radius of 50 cm. Note the logarithmic scale.

6.3 Cross-validation over test sets

The test set was chosen to cover a variety of conditions so that it would correspond to expected real-world performance. If the test set error is too high, then that means the training set was insufficiently diverse. If the test set error is too low, then either the training set is comprehensive or the test set contains only familiar configurations. An intermediate test set error would suggest the training set is somewhat diverse but the test set may contain unfamiliar or out-of-distribution configurations. To test the reliability of the test set, the one test set of 8 dataruns was hand chosen from dataset 04 (59 dataruns) was compared with iteratively generated test sets by randomly selecting remaining dataruns without replacement. The test set MSE performance can be seen in fig. 21. In summary, test set 0 (the hand-picked one) seems to be reasonably chosen since it does not lie at either extreme. The primary conclusion is that the current choice of test set 0 may give slightly optimistic predictions but should be a fairly reliable indicator of predictive performance. If the test set distribution were exponential, then the decay rate would be somewhere between 210 and 240. Looking at the z-scores, each test set appeared to be similarly calibrated with the exception of sets 3 and 5 which had very high ensembled MSEs and poorer calibration.

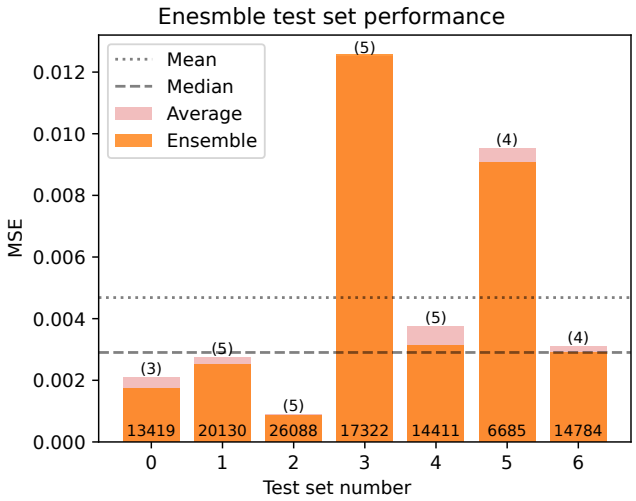


Figure 21: Test performance given the test set used. The ensemble error is calculated by first averaging the prediction and then calculating the MSE. The average error is calculated by averaging the MSE of each individual prediction. The median and mean of the MSE over the test sets (ensembled) are plotted. The numbers at the base of the bars indicate the number of shots in the particular set. The numbers in parentheses at the top of the bars indicate the number of training runs used in the calculation (some were excluded because training became unstable). test set 0 was hand picked; the remaining were randomly selected without replacement. Set 0 has a smaller error than the ensemble median but not by much, leading us to conclude that it is reasonably representative of the training data.

6.4 β-negative log likelihood loss

One can interpolate between an MSE loss and Guassian NLL loss by introducing an adaptive scaling factor StopGrad($\sigma_i^{2\beta}$) into the NLL loss function (eq. 2):

$$\mathcal{L}_{\beta-\text{NLL}} = \frac{1}{2}\left(\log \sigma_i^2(\mathbf{x}_n) + \frac{(\mu_i(\mathbf{x}_n) - y_n)^2}{\sigma_i^2(\mathbf{x}_n)}\right) \text{StopGrad}\left(\sigma_i^{2\beta}\right) \tag{3}$$

with an implicit expectation over training examples. $\beta = 0$ yields the original Gaussian NLL loss function and $\beta = 1$ yields the MSE loss function, so introducing a β factor can be interpreted as interpolating between these two loss functions. Thus, this factor improves MSE performance by scaling via an effective learning rate for each example (which is why StopGrad is used) [9], and may also improve both aleatoric and epistemic uncertainty quantification [10].

Using the β-NLL loss, the model is slightly better calibrated, but barely worth discussing. Adding a learning rate schedule, in this case one that's proportional to 1/epoch, improves test set error but worsens model calibration. The decrease in calibration performance could be because the model is more efficiently using its parameters so more weight decay could be needed. A representative example from the test set can be seen in fig. 22.

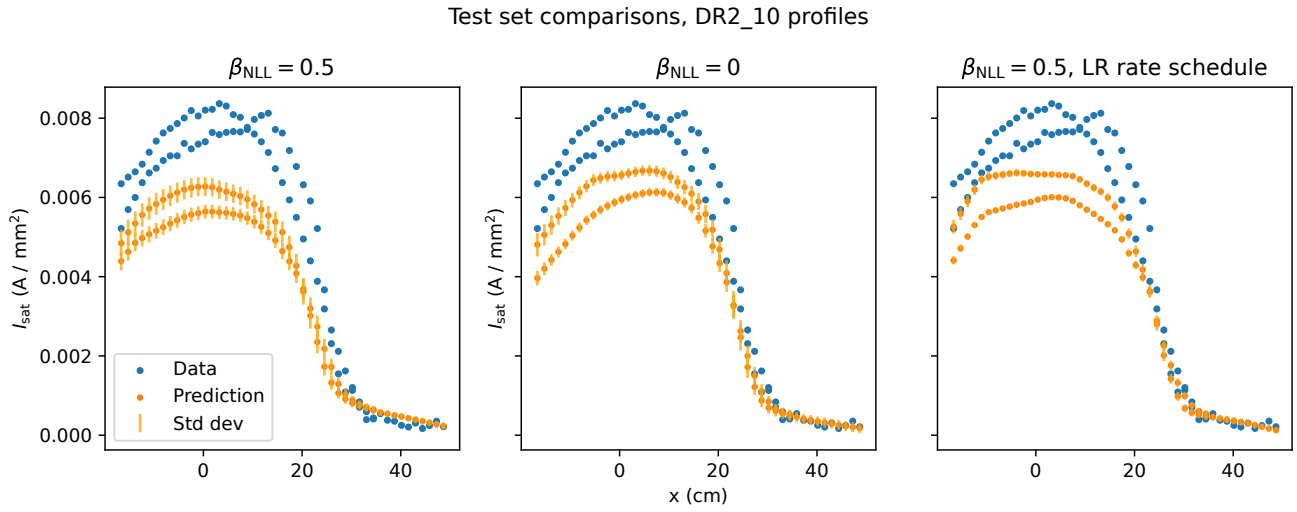


Figure 22: Test set performance, comparing models with and without β-NLL losses and a model trained with a learning rate decay schedule $\propto 1/\text{epoch}$.

6.5 Learning rate scheduling

Modifying the learning rate over time is known to improve model learning. I tried a few different learning rate decay methods: constant learning rate ($\gamma = 3 \times 10^{-4}$), $\gamma \propto \text{epoch}^{-1}$, $\gamma \propto \exp(\text{epoch})$, and $\gamma \propto \text{epoch}^{-1/2}$. The epoch is actually the training step divided by the number of batches in one epoch, so "epoch" in this case takes on a floating-point value. As seen in fig. 23, $\gamma \propto \text{epoch}^{-1}$ seems to give the best test set loss and everything beats a constant learning rate. The differences in test set losses are not too great, but the differences in training and validation losses are quite significant.

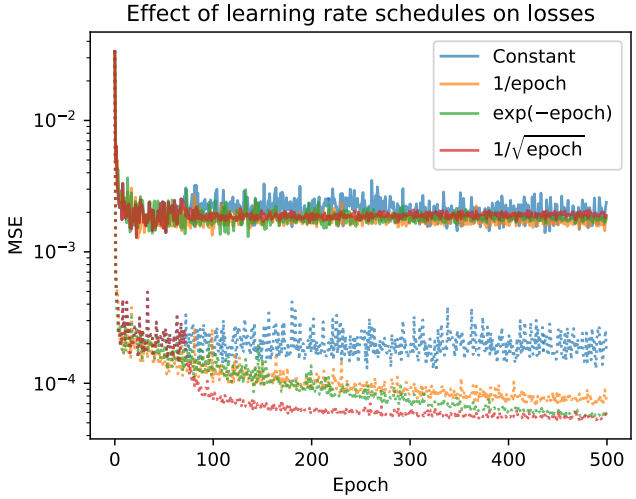


Figure 23: Different learning rates and the resulting test (top) and validation (bottom) set losses after 500 epochs. The $\gamma \propto \text{epoch}^{-1}$ schedule performs best on the test set, but not by much.

6.6 Combining β-NLL loss, LR scheduling, ensembles, and weight decay scans

We now combine the knowledge learned so far to get the best possible test set performance. Namely, we use the β-NLL loss with $\beta = 0.5$, schedule the learning rate (γ) where $\gamma \propto \text{epoch}^{-1}$, train 5 different models for an ensemble, and perform a scan over different weight decays. All this effort is so that we can find a good, calibrated model to make predictions with. A smaller network – 4 layers 512 wide – is used because it is faster to train, though it's important to point out that with 201,218 parameters it may not be sufficiently overparameterized for 118,131 examples.

The results of the weight decay scan can be seen in fig. 24. Increasing the weight decay increases the test MSE and decreases its z-score standard deviation. This increase in test MSE means that the model is making less accurate predictions, but the model is somewhat better calibrated. The model never becomes well-calibrated – the predicted uncertainty is always too low by a factor of 2 to 5. As seen in fig. 20, the uncertainty predictions are not useful except for models with low weight decays.

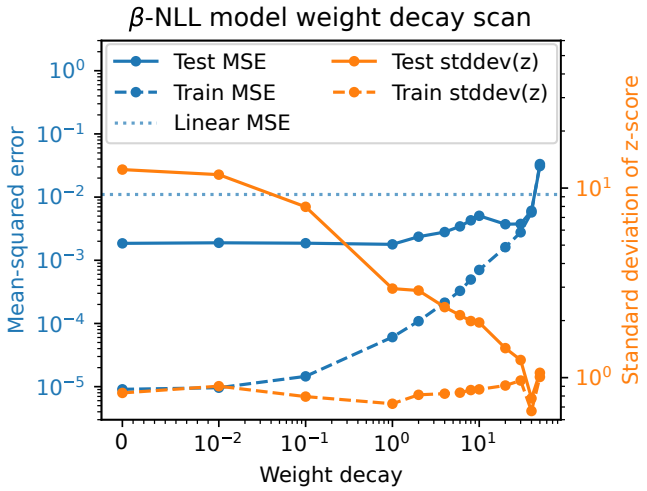


Figure 24: Model performance and calibration for different weight decays. Highly biased models are better calibrated, but come at great expense of mean prediction error. At the weight decay value of 50, the model has worse error than a linear model. Note the linear scale below 10^{-2}.

An example of extrapolating in the x-coordinate can be seen in fig. 25. The uncertainty beyond the training data is much better when the weight decay is smaller. A high weight decay coefficient constrains the variance of the network which includes regions outside of what is fit, so this makes sense. The variance predicted by the networks themselves are also high, indicting some other effects at play here with how weight decay interacts with training and generalization. This may be more clearly seen in fig. 26 with the error plotted below.

Calibrating the uncertainty estimate has proven to be challenging, though using the β-NLL loss function has provided benefits to training stability and generalizability. Further work is needed on uncertainty estimation.

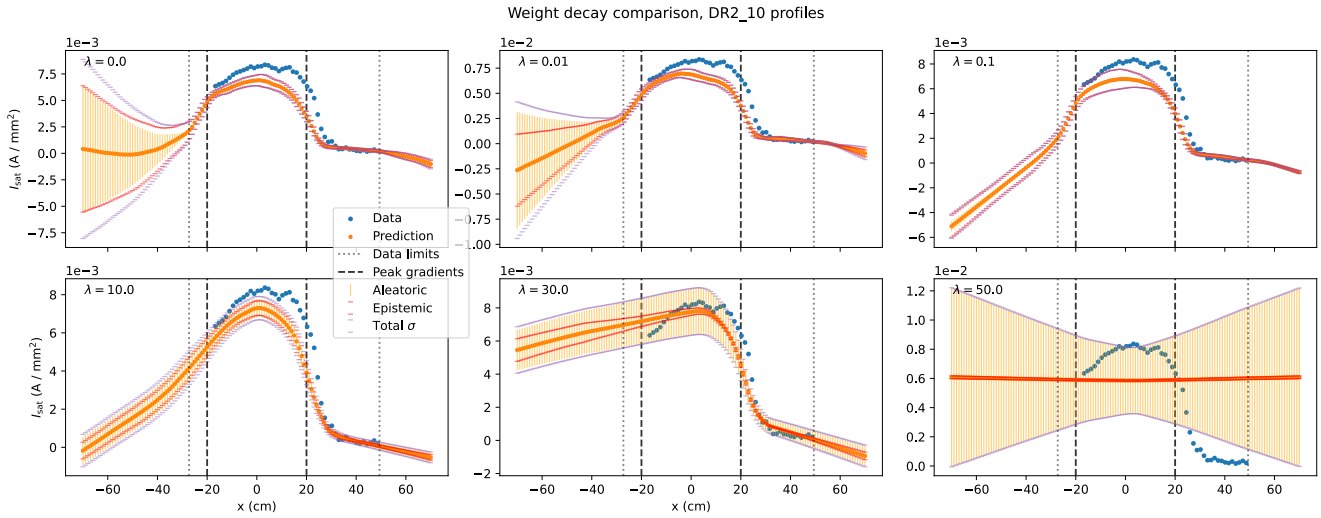


Figure 25: Model extrapolation performance with uncertainty for a model ensemble trained on a β-NLL loss function. Lower weight decay coefficients lead to more reasonable uncertainty measurements beyond the data range.

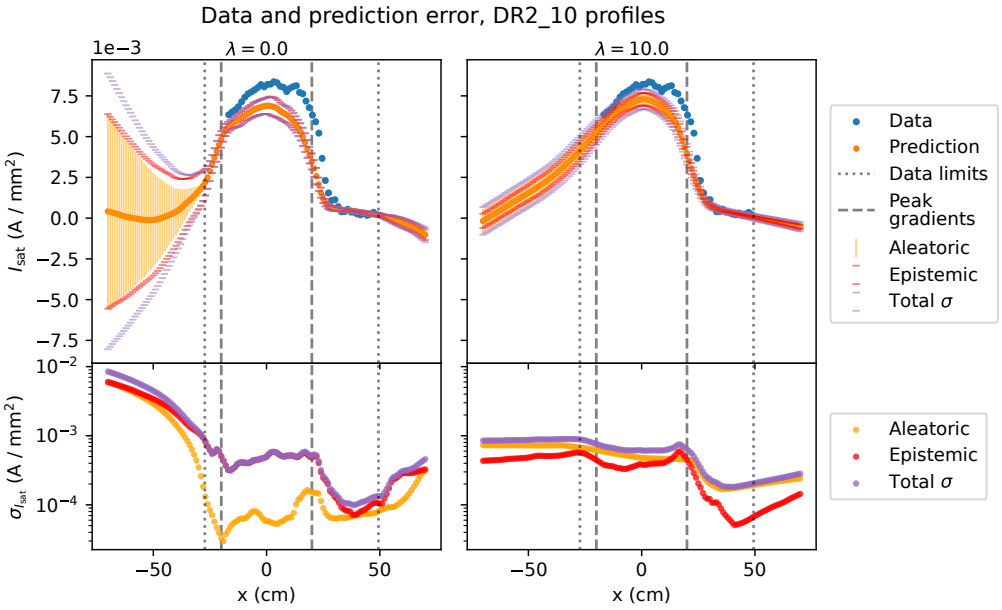


Figure 26: Model extrapolation performance with uncertainty for a model ensemble trained on a β-NLL loss function. The *relative* uncertainty appears to be more useful when zero weight decay is used: the uncertainty increases when the model is predicting outside its training data along the x-axis.

7 Validating the model

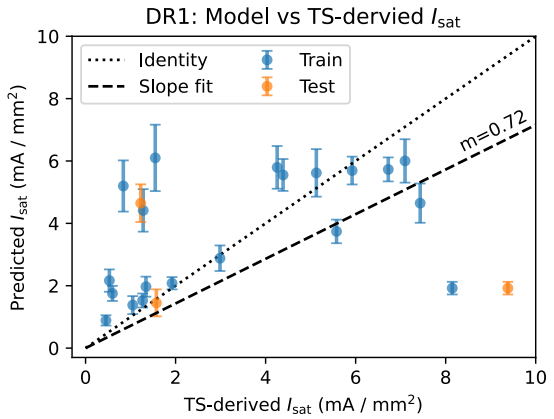
We validate the model in a couple different ways: seeing how the model predictions compare to simultaneous Thomson scattering measurements in the training and tests sets, and checking the

model against our intuition in mirror configurations.

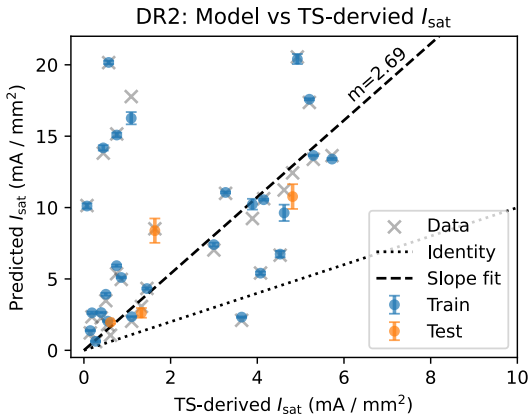
7.1 Thomson scattering: z-axis interpolation

The z-axis interpolation for dataruns in the training and test sets can be evaluated using the Thomson scattering (TS) diagnostic. The TS measurement is taken 8 ms into the discharge for `DR1` 12 ms into the discharge for `DR2`, but in this study the measured and predicted I_{sat} are instead averaged over 10 to 20 ms. The Thomson scattering measurement is compared in with `DR1` and `DR2` in figs. 27a and 27b, respectively. The linear slope fits do not take model error into account In `DR1`, I_{sat} predictions disagree with the I_{sat} derived from TS. Measurements from probes, when nearby the TS beam, can also have very different values from the TS-derived measurement. The TS density measurement may suffer from misalignment, and has not been calibrated since January 2022, roughly a year (`DR1`) or two (`DR2`) before these data were taken. The density measurement requires an absolute calibration because it is essentially photon counting. This disagreement likely comes from this error in density because fitting the temperature is robust to absolute calibration errors. In addition, the I_{sat} is time-varying; the average may differ substantially from single points in time earlier in the discharge (technically TS is a 4 ns average).

All these issues considered, the model predictions has rough agreement with TS on average in `DR1`, which is encouraging because the TS beam at port 32 (671 cm) is substantially further from the closest probe at port 27 (831). We should expect rough agreement (or perhaps a slight underestimate) on average because the model predictions have roughly symmetric error as see fig. 11). `DR2` has a probe past the TS beam at port 33 (639 cm), but the I_{sat} measurement rarely agrees with Thomson. Because of this density error and measurement time discrepancy, we conclude that the TS diagnostic may not be a good way to verify the predictions of the model. Note that, when calibrated, TS agrees with I_{sat} measurements quite closely as seen in the LAPD Thomson scattering paper [1].



(a) Thomson scattering (TS) 8 ms into the discharge compared to the model predictions (10 to 20 ms averaged). Broadly speaking, the TS measurement roughly agrees with the model estimate on average.



(b) Thomson scattering (TS) 12 ms into the discharge compared to the model predictions (10 to 20 ms averaged) and I_{sat} measurements one port away. The TS underestimates I_{sat} in general.

Figure 27

7.2 Checking intuition: modifying a M=3 mirror scenario

From geometric arguments (and experience), we know that modifying the mirror geometry can control the effective width of the plasma. One way to check that the machine model is learning appropriate trends is to check with this intuition. Namely, when the magnetic field at the source is not equal to the field at the probe, the probe will see the plasma expanded (or contracted) by roughly a factor of $\sqrt{B_{\text{probe}}/B_{source}}$. The cathode is about 35 cm in diameter, so a magnetic field ratio of 3 would give produce a plasma approximately 30 cm in radius. Given that all the probes are mostly in the midplane (or mirror cell) region, changing the cathode or midplane fields should change the measured width compared, but the mirror field may not have much of an effect.

To check this intuition, the model is fed with the following inputs: B_{source}=500 G, B_{mirror}=1500 G, B_{midplane}=500 G, discharge voltage=110 V, gas puff voltage=70 V, gas puff duration=38 ms, run set flag=1 and top gas puff=off. The discharge voltage and gas puffing parameters were arbitrarily chosen. The x coordinate is scanned from 0 to 30 cm, and the z coordinate from 640 to 1140 cm. This discharge is then modified by changing B_{source} to 1500 G, B_{midplane} to 1000 G (M=1.5), and B_{mirror} to 750 G. The model predictions are seen in fig. 28. Changing the source field to 1500 G increases the I_{sat} towards the edge of the plasma, as expected. When the midplane field is increased, the I_{sat} values further out decrease and increase at x=0 cm, which is consistent with a plasma column shrinking in width. When only the mirror field is modified, the strongest effect on I_{sat} is on or near x=0 cm, and the plasma column width does not appear to appreciably change. Interpretation of these predicted data are difficult because of the strong axial I_{sat} gradient present.

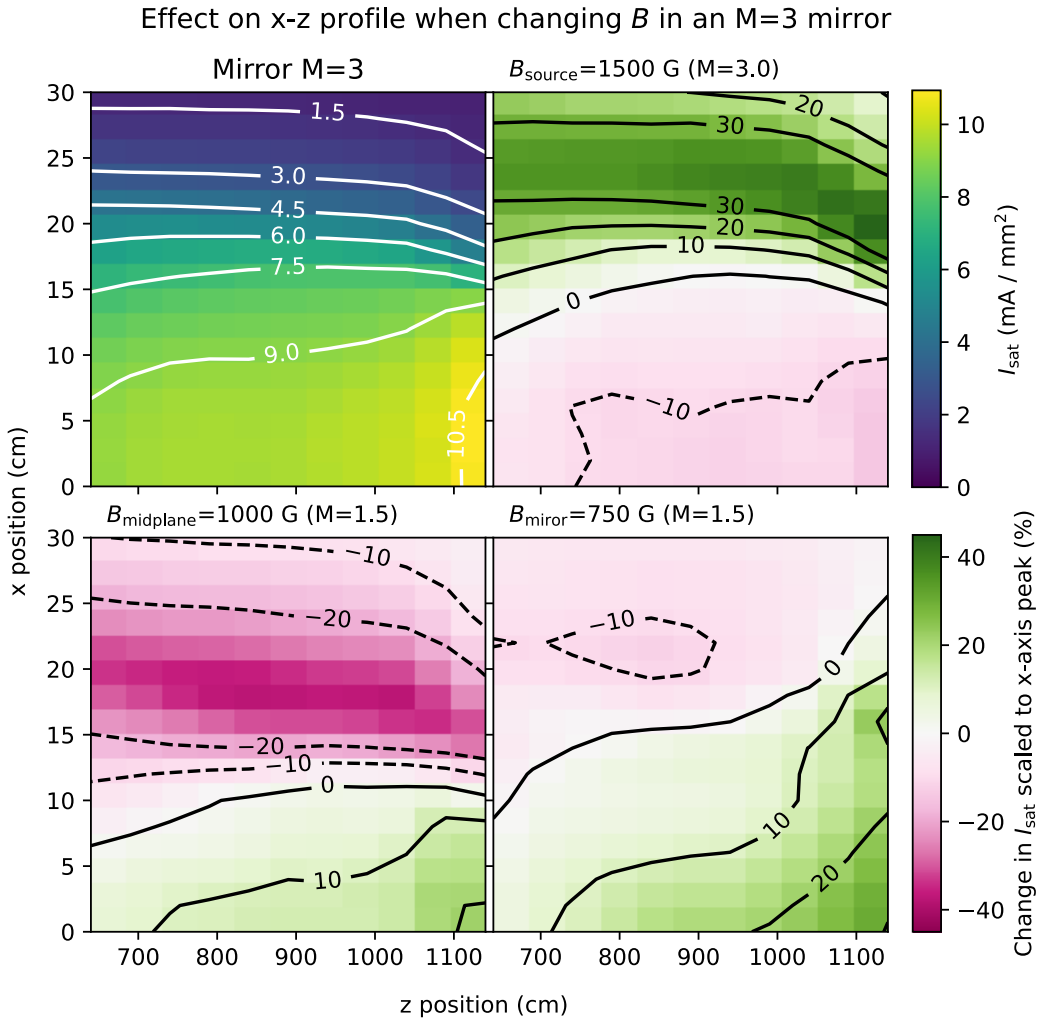


Figure 28: Top left: x-z I_{sat} profile. Other plots: relative change (percent) in I_{sat} when various fields are changed. The change in I_{sat} is normalized to the peak value on the x-axis so that changes in profile width can be more easily seen. Changing magnetic field clearly modifies the width of the plasma column in the mirror cell.

8 Physical insight via model inference

One item of particular interest is decreasing the axial density gradient in LAPD plasmas. Scientists may have good intuition on how the machine operates, but there is no systematic (or written) way of predicting what actuators will affect the axial gradient. The goal here is to predict what conditions lead to the flattest possible axial gradient in the LAPD.

For this analysis we use an ensemble of five β-NLL-loss models with weight decay $\lambda = 0$. The $\lambda = 0$ model is used because it appears to give the most useful uncertainty estimate as seen in fig. 25. The optimal machine actuator states are found by feeding a grid of inputs into the neural network. This variance estimate is not well-calibrated: the error of the predictions on the test set falls far outside the predicted uncertainty. However, this uncertainty can be used in a relative way: when the model is predicting far outside its training range, the predicted variance is much larger.

The ranges of inputs into this model are seen in table. 4. These inputs yield 127,234,800 different machine states (times five models) which takes 151 seconds to process on an RTX 3090 ($\approx$ 4.2 million forward passes per second) when implemented in a naive way. Note that gradient-based methods can be used here instead because the network is differentiable everywhere but this network is sufficiently small that a comprehensive search is tractable.

Table 4: Machine inputs and actuators for model inference

Input or actuator	Range	Step	Count
Source field	500 G to 2000 G	250 G	7
Mirror field	250 G to 1500 G	250 G	6
Midplane field	250 G to 1500 G	250 G	6
Gas puff voltage	70 V to 90 V	5 V	5
Discharge voltage	70 V to 150 V	10 V	9
Gas puff duration	5 ms to 38 ms	8.25 ms	5
Probe x positions	-50 cm to 50 cm	2 cm	51
Probe y positions	0 cm	–	–
Probe z positions	640 cm to 1140 cm	50 cm	11
Probe angle	0 rad	–	–
Run set flag	off and on	1	2
Top gas puff flag	off and on	1	2

8.1 Minimal and strongest $I_{\mathbf{sat}}$ axial variation

One particular issue seen in LAPD plasmas is sharp axial density and temperature gradients in the machine. The required LAPD state for attain the flattest possible axial profile can be found by finding the minimum standard deviation along the z-axis at $x, y = 0$:

$$\text{Inputs} = \underset{\text{Inputs}\neq z}{\arg\min}\, \underset{z}{\text{sd}}(I_{\text{sat}}|_{x=0}) \tag{4}$$

This minimization disproportionately penalizes outliers. Like any optimization method, the results may be pathologically optimal. In this scenario, the minimal axial variation is found when the I_{sat} is only around 1 mA/m^2, which is quite small and corresponds to 1-2$\times 10^{12}$ cm^{-3} depending on Te. As seen in fig. 29, the predicted uncertainty and test set RMSE are very large in comparison, which indicates the model may be far outside the bounds of the training data. The predicted aleatoric uncertainty is also quite large, suggesting that the measured I_{sat} values may fluctuate considerably.

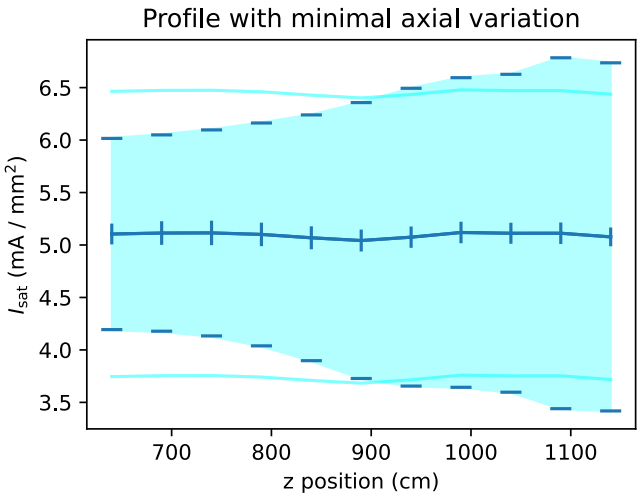


Figure 29: Flattest profile as measured by the standard deviation along the z-axis within the machine states indicated in table 4. The predicted error is quite large – even larger than the test set RMSE – which indicates the model is may be outside the bounds of the training data.

Since many physics studies require higher densities, we constrain the mean axial I_{sat} value to be greater than 7.5 mA/mm^2 (roughly 0.5-2 $\times 10^{13}$ cm^{-3}). We also consider what machine settings would lead to the greatest axial variation out of curiosity. The results of both of these optimizations can be seen in fig. 30. The optimizations yield profiles that have the largest I_{sat} values closest to the cathode, which is expected.

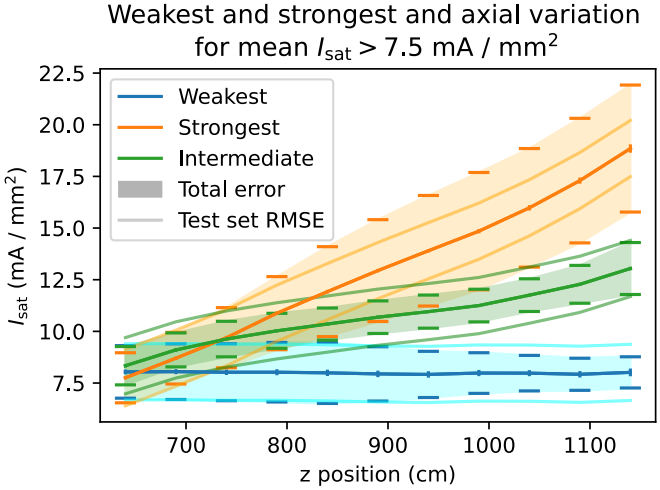


Figure 30: Strongest and weakest axial variation when I_{sat} is constrained to be greater than 7.5 mA/mm^2. The predicted error for the strongest axial variation is much smaller than for the weakest case, so the large gradient is more likely to be found in the data than the small one. The model was also constrained to find conditions where the top gas puff was off and the machine condition run flag was 1. An intermediate case was also found to add another test point for trend evaluation.

The required LAPD machine state for these optimizations are listed in table. 5. The "run

set flag" is set to "on" for cases to be validated (bolded) because we do not want to turn off the turbopumps to increase neutral density. The difference between the top puff valve off or on for strongly varying axial profiles is on average -2 mA/mm^2 but otherwise the two have very similar shapes.

Table 5: Machine inputs and actuators for optimized axial profiles

Input or actuator	Weakest I_{sat} = any	**Weakest** $I_{\text{sat}} > 7.5$	Strongest $I_{\text{sat}} > 7.5$	**Strongest** $I_{\text{sat}} > 7.5$
Source field	750 G	1000 G	1250 G	500 G
Mirror field	1000 G	750 G	750 G	500 G
Midplane field	250 G	250 G	1250 G	1500 G
Gas puff voltage	70 V	75 V	85 V	90 V
Discharge voltage	130 V	150 V	150 V	150 V
Gas puff duration	5 ms	5 ms	38 ms	38 ms
Run set flag	on	on	off	on
Top gas puff flag	on	off	off	off

8.2 Validating prediction for strongest axial variation

Data were collected on September 18th, 2024 to validate the predictions made for the strongest axial variation. Given the model inputs were incorrect (the gas puff voltage was swapped with discharge voltage), this optimization is broadly wrong. However, because the prescribed gas puff voltage and discharge voltage were the same (90 V) and within the bounds of the actuator training data, we can still use these data for validation. The parameters used were B_{source}=1250 G, B_{mirror}=500 G, B_{midplane}=1500 G, discharge voltage=90 V, gas puff voltage=90 V, gas puff duration=38 ms, run set flag=1 and top gas puff=off.

Only a single useful shot was collected at a nominal -45°angle 10 cm past the center (x=0 cm, y=0 cm) of the plasma on three probes at z-positions of 990, 767, and 607 cm (ports 22, 29, and 34, respectively). These probes were also not centered which so the positions were slightly further +x and -y. Issues (scripts crashing) with the probe drives and oscilloscope readout resulted in only a single shot being recorded. The resulting predictions using these coordinates and machine conditions can be seen in fig. 31. The off-axis predictions and I_{sat} reading result in less-smooth profile when compared with the on-axis prediction seen in fig. 30. The model seems to reproduce the trend well, but underestimates I_{sat} on an absolute comparison. Probe tip rotation was not accounted for in the prediction, but including the correct rotation produces uncertainty meeting or exceeding the test RMSE threshold because rotations at this angle have not been seen by the model. This probe tip rotation bias suggests that for this model probe rotation need not be correct for accurate predictions. The model performs worse when azimuthal symmetry is assumed by treating x as the radial coordinate and calculating the distance from x=y=0 cm. This degraded inference performance, also seen when training the model (section 5.8) suggests that there is some intrinsic azimuthal asymmetry in the plasma, possibly caused by the horizontal gas puff injection near the source.

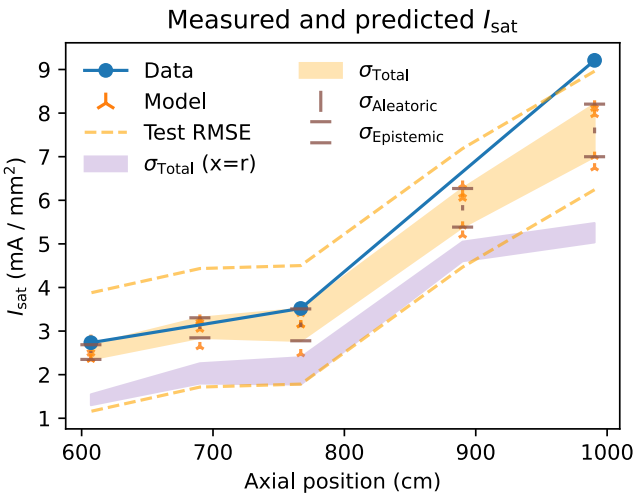


Figure 31: Data collected at off-axis positions around x= 9.75 cm and y= −8.4 cm are compared with predictions from the machine learning model at the same points in addition to two interpolating predictions. The model predicts the trend well, but underestimates I_{sat} in general. The shaded orange region is the total model uncertainty ($\sigma = \sqrt{\text{Var}}$). The shaded purple region is the total uncertainty of the model prediction when instead the x coordinate is used as a radial coordinate, i.e., assuming azimuthal symmetry of the plasma.

This small run had several issues, one of which was that the cathode was in a odd and unexpected state state because the heater was not returned to a more typical operating range, and thus the cathode was not in a typical operating regime, before my shots were taken. This odd state resulted in the discharge current being significantly lower than expected when compared with past (human) experience. In addition, the bank voltage supply was in the constant-current, not constant-voltage mode. I do not know how the supply mode affects the discharge.

8.3 Validating strongest, weakest, and intermediate axial variation

The predictions for the strongest, weakest, and intermediate axil variation cases is seen in fig. 30. The intermediate case was chosen as somewhere around half way between the strongest and weakest case with a round index number (15000, in this case). The intermediate case has the following machine configuration: B_{source}=2000 G, B_{mirror}=1250 G, B_{midplane}=750 G, gas puff voltage=90 V, discharge voltage=120 V, gas puff duration=38 ms, run set flag=1 and top gas puff=off.

Prediction figure reference
Comparison with measured data
Discrepancy from Langmuir probe uncalibration
Attempted correction of measured data
Interferometer traces, temperature measurements, experience for past 5 ms puffing runs
Centering probes using FFC

8.4 Extrapolating to a higher discharge voltage

While waiting the cathode to return to normal opearating condition, I took a few 45° line measurements. The operating regime was an odd one: $B_{\textbf{source}}$**=822 G**, B_{mirror}=500 G, B_{midplane}=1500 G, **discharge voltage=160 V**, gas puff voltage=90 V, gas puff duration=38 ms (and run set flag=1

and top gas puff=off for model inference). When plugged into the model, we will be interpolating in the source magnetic field and extrapolating in the discharge current. The result of this prediction (nearly) on-axis can be seen in fig. 32.

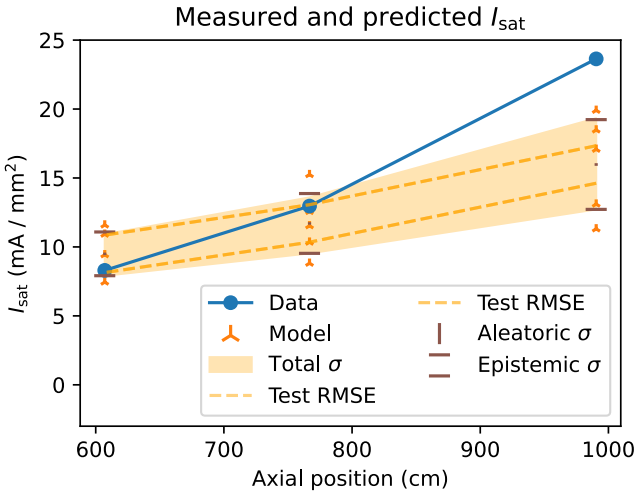


Figure 32: Measured vs predicted I_{sat} values for an odd machine configuration with B_{source}=822 G and discharge voltage=160 V. The machine was also in an odd discharge state, so it's no surprise that the predicted uncertainty bounds are very large (even greater than the test set RMSE value) and that accuracy suffers.

8.5 Correlation of gas puff duration with axial gradients

When initially performing the experiments to collect this dataset I did not expect the gas puff duration to be such a large actuator. With the shorter 10 and 5 ms gas puff durations the I_{sat} gradient appeared to decrease substantially. To verify this observation, we select discharges predicted from the model with a flat 1 kG field with the probe in the center. The discharge voltage was set at 110 (no reason, just a middle value) with the run-set flag at 1 and no top gas puffing. The plotted trend inferred from the model can be seen in fig. 33. Care was taken to handle the aleatoric (independent) error separately from the axially-dependent epistemic error.

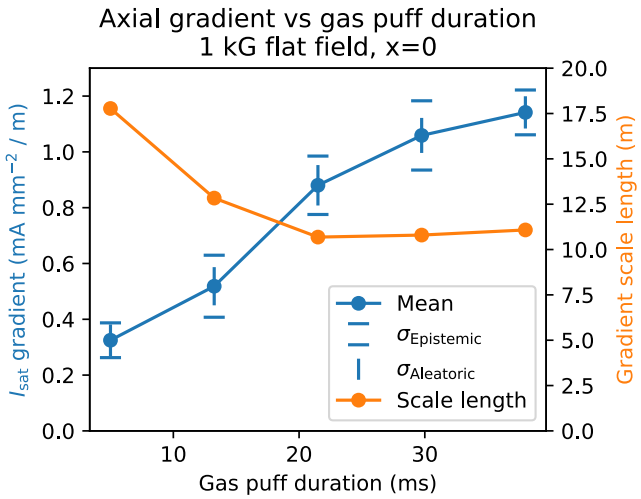


Figure 33: ML prediction: mean axial gradients decrease with decreased gas puff duration. Five durations are plotted between 5 and 38 ms (which are the bounds of the training data), evenly spaced. The gradient scale length also increases, indicating that the gradient change was not just from a decrease in the bulk I_{sat}.

As seen in the figure, the mean axial gradient decreases when the gas puff duration is shortened. The gradient scale length also increases, so the mean gradient is not decreasing simply because the bulk I_{sat} is decreasing. This result confirms expectations and may be a useful actuator to consider when planning future experiments.

8.6 Effect of changing the discharge voltage

I do not have a good intuition on what changing the discharge voltage actually does, so I collected a bunch of predictions from the model to see what would happen. Model parameters were chosen to be reasonable values: 1 kG flat field, 80 V gas puff, 38 ms gas puff duration, run set=1, and top gas puff off. The 38 ms puff is chosen in a lot of these predictions because it is the most common gas puff duration by far (see table 2). The results of changing the discharge voltage only can be seen in fig 34. A reduced I_{sat} would be consistent with an increased temperature and a decreased density. Unfortunately the discharge current was not included in the training set (as an output), otherwise discharge power could be calculated and directly compared with measurements. As discussed in Ghazaryan et al. [1], at constant gas inlet pressure $T_e \sim P^2$, where $P = IV$ is the discharge power. Since $n_e T_e \sim P$ (assuming constant losses) and $I_{\text{sat}} \sim n_e \sqrt{T_e}$, this implies that $n_e \sim 1/P$ and thus $I_{\text{sat}} \sim 1$, so this doesn't make any sense.

We expect this model to be more accurate at 150 V discharge voltages because the model is likely more biased in that region of parameter space as seen in table 2.

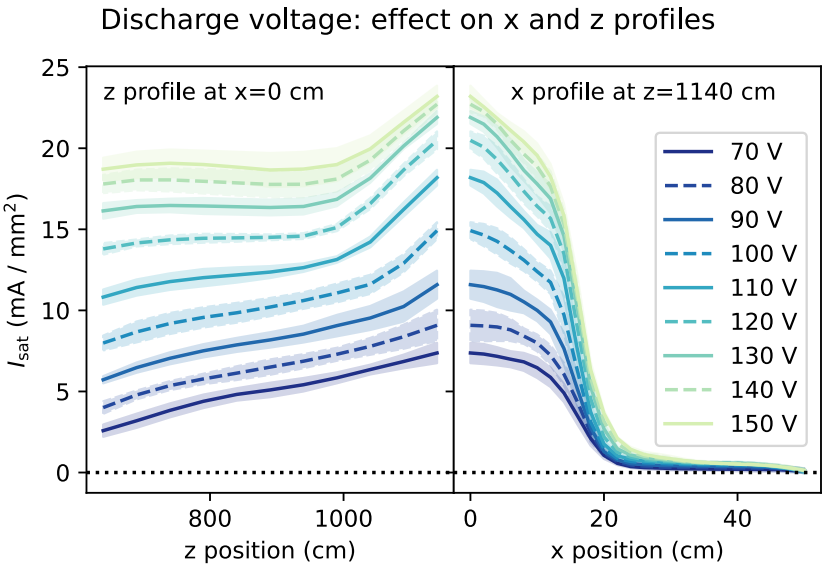


Figure 34: The z profile at x=0 cm and x profile at z=1140 cm for different discharge voltages. The I_{sat} decreases with increasing voltage, and the error (filled regions) stays roughly the same, but in general increase slightly towards the cathode and at higher discharge voltages.

8.7 Azimuthal asymmetry

The lack of azimuthal symmetry of the model, as hinted in training (sec. 5.8) and inference (sec. 8.2) can be seen in a 2d histogram of I_{sat} at points along either the x or y axis corresponding to the same distance from the axis of the device, shown in fig. 35. Positions between 0 and 20 cm (1 cm spacing) for x and y are plotted because y=20cm is at the limit of probe movement in this dataset, but the radius limit is probably further because of the x-y hypotenuse. These results should be treated with a degree of skepticism because planar data lacks diversity in this dataset; most dataruns are simply y=0 lines.

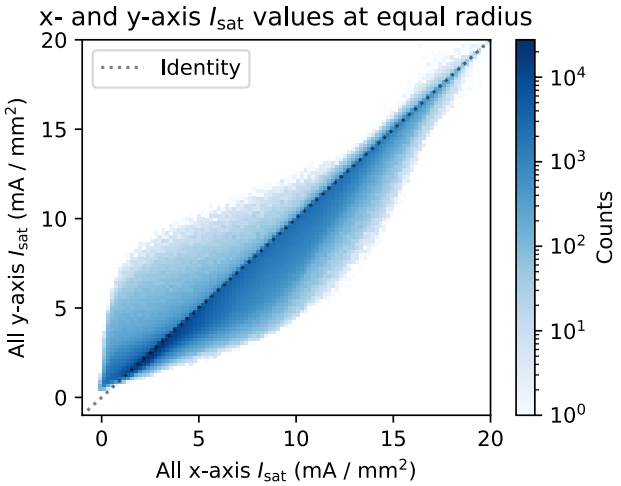


Figure 35: I_{sat} along the x- and y-axis plotted against each other for identical positions. Points off the identity line indicate some x-y (or azimuthal) asymmetry.

9 Issues with this work and methodology

- We do not enough training data to reliably predict the absolute I_{sat} values.
- I_{sat} suffers from absolute calibration issues
- The output is very limited – it's just a time-averaged single quantity – so expanding this to other measurements could be really cool.
- Network is small enough and the problem is simple enough for an exhaustive search for optimization purposes. This simplicity and cheapness will not hold true for future models, so another method will be needed.

10 Future work

10.1 Collecting more validation data

I only got one useful validation shot. I need to collect data with the three non-top-puff dataruns in table 5. Once is chance, twice is luck, thrice is skill.

10.2 Adding non-actuator inputs

Including information that the plasma provides (instead of just the machine settings) could be useful, both for prediction accuracy (though in this case I would call it "inference", not "prediction") and for determining useful non-physically interpreted signals (e.g., diodes, fast framing camera).

10.3 Generative modeling

Learning a generative model (like an energy-based model) could be really useful because we get inversion for free in any parameter. The generative modeling may also provide a greater general-

ization capability (maybe?) because the model needs to learn all correlations instead of just the output.

10.4 Adding in time series information

If the dataset is sufficiently diverse, eventually we want to predict time series of profile evolution. This will be a precursor to fluctuation predictions.

10.5 Turbulence and transport

If possible, predicting fluctuations, spectra, and particle flux should at least be attempted. Optimizing transport (or general fluctuation level) in a device like the LAPD could be a cool proof of concept for extending this to a fusion reactor.

10.6 Simply gathering more data

Adding additional, diverse data would improve performance of the model and increase our confidence in its predictions. More x-y plane data collection would also help quantify azimuthal asymmetry and potential reasons for its occurrence.

References

[1] S. Ghazaryan, M. Kaloyan, W. Gekelman, Z. Lucky, S. Vincena, S. K. P. Tripathi, P. Pribyl, and C. Niemann. Thomson scattering on the large plasma device. *Review of Scientific Instruments*, 93(8):083514, August 2022.

[2] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On Calibration of Modern Neural Networks. http://arxiv.org/abs/1706.04599, August 2017. arXiv:1706.04599 [cs].

[3] Andrej Karpathy. A recipe for training neural networks. https://web.archive.org/web/20240709000647/http://karpathy.github.io/2019/04/25/recipe/, April 2019. Accessed: 2024-07-12.

[4] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. http://arxiv.org/abs/1612.01474, November 2017. arXiv:1612.01474 [cs, stat].

[5] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep Double Descent: Where Bigger Models and More Data Hurt. http://arxiv.org/abs/1912.02292, December 2019. arXiv:1912.02292 [cs, stat].

[6] D.A. Nix and A.S. Weigend. Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, pages 55–60 vol.1, Orlando, FL, USA, 1994. IEEE.

[7] Rylan Schaeffer, Mikail Khona, Zachary Robertson, Akhilan Boopathy, Kateryna Pistunova, Jason W. Rocks, Ila Rani Fiete, and Oluwasanmi Koyejo. Double Descent Demystified: Identifying, Interpreting & Ablating the Sources of a Deep Learning Puzzle. http://arxiv.org/abs/2303.14151, March 2023. arXiv:2303.14151 [cs, stat].

[8] Prem Seetharaman, Gordon Wichern, Bryan Pardo, and Jonathan Le Roux. AutoClip: Adaptive Gradient Clipping for Source Separation Networks. http://arxiv.org/abs/2007.14469, July 2020. arXiv:2007.14469 [cs, eess, stat].

[9] Maximilian Seitzer, Arash Tavakoli, Dimitrije Antic, and Georg Martius. On the Pitfalls of Heteroscedastic Uncertainty Estimation with Probabilistic Neural Networks. http://arxiv.org/abs/2203.09168, April 2022. arXiv:2203.09168 [cs, stat].

[10] Matias Valdenegro-Toro and Daniel Saromo. A Deeper Look into Aleatoric and Epistemic Uncertainty Disentanglement. 2022. arXiv:2204.09308 [cs.LG].